NOVEL MOLECULAR COMPUTATIONAL METHODS AND THEIR QUANTITATIVE ASSESSMENT

by

Xin Zhang

Bachelor of Science, Nanjing University, 2004

Master of Science, University of Pittsburgh, 2006

Submitted to the Graduate Faculty of

the Department of Physics & Astronomy in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2010

UNIVERSITY OF PITTSBURGH

DEPARTMENT OF PHYSICS & ASTRONOMY

This dissertation was presented

by

Xin Zhang

It was defended on

July 19th 2010

and approved by

David Jasnow, Department of Physics & Astronomy

Daniel M Zuckerman, Department of Computational & Systems Biology

Anthony H. Duncan, Department of Physics & Astronomy

Jeremy Levy, Department of Physics & Astronomy

Hagai Meirovitch, Department of Computational & Systems Biology

Vladimir Savinov, Department of Physics & Astronomy

Dissertation Advisors: David Jasnow, Department of Physics & Astronomy,

Daniel M Zuckerman, Department of Computational & Systems Biology

NOVEL MOLECULAR COMPUTATIONAL METHODS AND THEIR QUANTITATIVE ASSESSMENT

Xin Zhang, PhD

University of Pittsburgh, 2010

Molecular computational methods and means for assessing their efficiency are discussed in this thesis. Computer simulations of biomolecules help to understand fundamental biological processes, as well as aiding drug design and many other crucial applications. Several efforts to improve biomolecular simulations are described in this thesis. First, a new algorithm based on polymer growth strategies is introduced. The main novel feature of this approach is the use of pre-calculated statistical libraries of molecular fragments. A molecule is sampled by combining fragment configurations of single residues, which are stored in the libraries. This method is demonstrated to be accurate and can generate configurational ensembles for large peptides (*i.e.,* 16 residues) in less than a minute of single-processor computing. As an application of this growth algorithm, a practical method is developed to calculate absolute free energy that stages such calculation in several steps through growing a molecule. Significant computer time is saved by pre-calculating fragment configurations and interactions for re-use in a variety of molecules. To assess the growth method and other approaches, the question "how much faster is a method than standard molecular dynamics?" is addressed. A general method for the assessment of sampling quality is needed to quantify the progress in the development of algorithms and forcefields used in molecular simulations. I therefore develop an approach for analyzing the variances in state populations, which quantifies the degree of sampling in terms of the effective sample size (ESS). This procedure is tested in a variety of systems from toy models to atomistic protein simulations. Lastly, a simple automated procedure is introduced to obtain approximate physical states from dynamic

trajectories: this allows sample size estimation in systems for which physical states are not known in advance.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

A. CONTRIBUTION OF PHYSICS IN STUDYING BIOLOGICAL SYSTEMS

The study of biological systems at the molecular level focuses on the molecular structures associated with their functions. In addition, there are always fluctuations. From small peptides to large proteins, there are big conformational changes in their motions. If proteins stay frozen in unique structures, like those tabulated in the Protein Data Bank [1] (a database for the 3-D structural data of large biological molecules), nothing will happen, *i.e.,* no life exists. So at any finite temperature, proteins have ensembles of configurations, which are Boltzmann-distributed in equilibrium. In other words, proteins are more like machines that function by motions, and statistical ensembles of protein configurations are the simplest way to describe such motion. That is why physics has made a huge contribution in biological systems investigation [2].

During the last decades the well-established tools of statistical physics have been extensively applied to a rising number of biological phenomena. In simple terms, statistical physicists have been able to establish the theory for physical systems consisting of a large number of interacting particles, such as occurs in biological systems [3]. The origin of fluctuations in biomolecules is usually thermal noise or erratic motion of microscopic particles. In the "thermal bath" surrounding proteins, there are energy exchanges, "kicks" from stochastic forces. All these phenomena have been studied in physics.

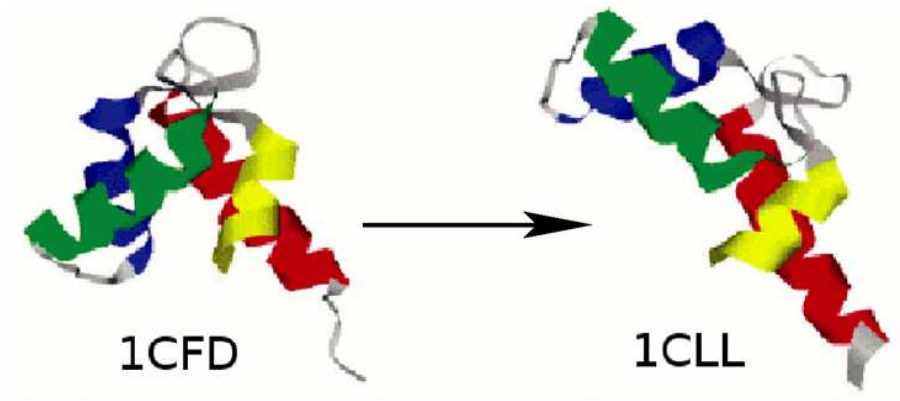

Figure 1: Ribbon representation of the N-terminal domain of the protein calmodulin. The left panel is the calcium-free (apo) structure and the right panel depicts the calcium-bound (holo) structure. Ions are not shown.

What people need from statistical physics is mostly the quantities that could be calculated from equilibrium ensemble. For instance, knowledge of free energy for two different states or systems of interest allows the estimation of stabilities and determines binding affinities of ligands to proteins [4, 5]. Researchers could also calculate average end-to-end distance from the ensemble, which could be measured and compared in experiments [6].

Bio-molecules often undergo dramatic conformational change during their dynamic processes [7]. Fig. 1 is an example of the protein calmodulin that undergoes conformational change from calcium-free structure to calcium-bound structure. In a simulation, which is usually under the canonical condition (constant number of molecules, constant volume and constant temperature), this dynamic process will converge into equilibrium given enough steps. What I have studied is all about equilibrium ensemble. But it is very challenging to collect valid statistical sampling of Boltzmann-distributed configurations due to the complex energy landscape in large systems. I will detail this part in Chapter II. Once one gets the correct ensemble, he or she could find the dominant configurations in certain systems and calculate most physical quantities in equilibrium.

B. COMPUTER SIMULATION OF BIOMOLECULES

Biological systems are explained and studied by statistical mechanics. But one cannot derive everything by hand due to the high number of degrees of freedom in biological systems. While experiments are useful in determining many properties of biological processes, they are very limited in quantitatively studying physical properties at molecular level. For instance, the fully detailed dynamics of protein folding is not observable and the entropy of the systems is not measurable in experiments. An advantage of computer simulations over traditional experimental methods is the researcher has complete control over every aspect of the simulations [8], which are usually fast and cheap to perform and are easy to reproduce. The models can be formulated so that they obey fundamental laws such as conservation of mass and energy, constant temperature or number of molecules. Thus computational methods are increasingly becoming accepted as a complementary technique to experiments. The rise in computer power and improvement in theoretical algorithms enabled computational simulations to investigate the physical properties of structure and function of biological macromolecules. This section provides an brief introduction of some frequently used procedures for investigating protein function and dynamics under various conditions. I will also give their limitations later in this section.

1. Molecular modelling

Molecular modelling is a general term for theoretical methods and computational techniques that are applied to mimic the behaviour and motion of molecules, ranging from small peptide systems to large biological molecules.

Current computer simulation techniques can be classified into several categories based on the studied system size and time-scale capabilities. The most accurate methods are based on quantum mechanics (QM), usually by Density Functional Theory (DFT) methods [9] that considers the electronic structure, which is computationally prohibitive for proteins. So protein models are too large to be treated by these techniques. However, the QM methods are mostly used for investigating processes which involve electronic rearrangement.

Larger system sizes and longer simulation time scales can be investigated using the atomistic forcefield methods or "molecular mechanics" [10]. Molecular mechanics is applied when physical interactions are considered and based upon a classical model of interactions within a system with contributions from dynamic motion. Such motions may include bonds stretching, angle bending and rotations about single bonds. This method can provide accurate calculation of physical phenomena such as protein dynamics. All the calculations performed in this thesis are in the use of this method with proper forcefield, which refers to the functional form and parameter sets used to describe the potential energy of a system. Detailed discussion of force fields behind this powerful technique is presented later in this chapter at I.B.2.

Additionally, "coarse-grained models" are receiving significant attention due to their ability to simulate large complexes at timescales currently inaccessible to atomistic methods [11, 12]. These methods can also provide qualitatively accurate representation of the macroscopic states and some properties of a system over long timescales. In coarse-grained models a small group of atoms is treated as a single unit or "bead", where the dynamics of the system of beads is governed by a very simple forcefield. There is an intrinsic difficulty in the parameterisation of coarse-grained forcefields, related to the fact that complex and diverse interactions must be described by a small number of parameters. However, continued development and improvements of different coarse-grained models means this method has great potential for modelling large complex systems comparable with experiments.

Please note that, in most of this thesis, molecular mechanics or atomistic forcefield methods will be used on molecules amenable to this approach with current computational power, and I analyze some coarse-grained simulation data on Chapter IV.

2. Forcefield

Most molecular modelling, such as molecular dynamics and Monte Carlo, involves energy calculation. The potential energy calculation is derived from potential energy functions, called a "forcefield". The most commonly used forcefields for biomolecular simulations are the all-atom CHARMM [13], AMBER [14], OPLS-AA forcefields [15], where all the atoms

in the proteins are represented explicitly. There are also the united-atom GROMOS [16], OPLS-UA forcefields, where the hydrogen bonded to a carbon is treated as a single bead, thus named "united-atom". While different forcefields may have different parameters, the potential functions are pretty much the same. A brief introduction to these energy functional forms and parameterisation is discussed in the following.

The underlying functional forms of each of these forcefields contains energy terms describing the bonded (U_{bonded}) and nonbonded ($U_{\text{nonbonded}}$) interactions between the atoms of a system. The terms representing bonded interactions account for the stretching of bonds, the bending of angles, and the rotation of dihedrals. The electrostatic and van der Waals interactions are defined as nonbonded interactions. The bonded interactions are typically defined as

$$U_{\text{bonded}} = \sum_{bonds} k_b(b - b_0)^2 + \sum_{angles} k_\theta(\theta - \theta_0)^2 + \sum_{dihedrals} k_\chi(1 + \cos(n\chi - \sigma)) \qquad \text{(I.1)}$$

The first term in Eq.(I.1) describes the stretching of bonds in a quadratic form, equivalent to that of Hooke's law for the potential energy of a spring. The equation is a sum over all bonded pairs of atoms, where b is the bond length, k_b and b_0 are the parameters describing the stiffness and the equilibrium length of the bond, respectively. Determination of parameters will be discussed later this section. Please note that the bond length b could be any positive value. The shape of potential energy curve for different type of bonds are slightly different but very similar. The Morse potential [17] is a very useful functional form that models the curves and it describes a wide range of behaviour from strong equilibrium to dissociation. In molecular mechanics calculations, it is very rare for bonds to deviate significantly from their equilibrium values. Thus Hooke's law formula, an approximate to Morse potential, is picked as forcefield potential function. The second term describes the bending of the angle. k_θ and θ_0 are the parameters describing the stiffness and equilibrium position of the angle, respectively. The bond angle θ ranges from 0 to 180 degrees, often close to θ_0. Similar to the bond stretching term, this equation also has a quadratic form, which ignores higher order terms. The third term from the bonded interaction energies equation describes the energetics associated with rotation of the dihedral angle defined by quadruplets of consecutively bonded

atoms. As dihedral rotations are periodic in nature, a cosine function is used, where χ is the value of the dihedral, k_χ is the energetic parameter that determines the barrier height, n is the periodicity and σ is the phase. The bond length could be any positive value, although it is often approximate to b_0. There is no restriction on angles, thus the potential energy calculation covers the whole configuration space.

The nonbonded energies are calculated as

$$U_{\text{nonbonded}} = \sum_{nonbonded} \left\{ \epsilon_{ij} \left[\left(\frac{R_{min,ij}}{r_{ij}} \right)^{12} - 2 \left(\frac{R_{min,ij}}{r_{ij}} \right)^{6} \right] + k_e \frac{q_i q_j}{r_{ij}} \right\} \tag{I.2}$$

In the four widely used forcefields mentioned above, nonbonded interactions between atoms are defined as occurring either between atoms in separate molecules or between atoms separated by two or more bonds in the same molecule. The first part of this equation is the van der Waals term, in the form of a Lennard-Jones potential. The prefactor ϵ_{ij} is a energy parameter based on the types of the two interacting atoms i and j. $R_{min,ij}$ is a parameter that also depends on the types of the two interacting atoms and is close to the distance at which the Lennard-Jones energy is minimum. The second part of this equation models the electrostatic interactions between nonbonded pairs of atoms and is based on Coulomb's law. As with the Lennard-Jones equations, r_{ij} is the interatomic distance, while q_i and q_j are the parameters that describe the effective charges on atoms i and j and k_e is Coulomb constant. The total energy is defined as

$$U_{\text{total}} = U_{\text{bonded}} + U_{\text{nonbonded}} + U_{\text{other}} \tag{I.3}$$

where U_{bonded} is the contribution to the total energy from the bonded interactions, $U_{\text{nonbonded}}$ is the contribution from the nonbonded interactions, and U_{other} is mainly from the improper dihedral angles, which will not be discussed in detail here. Please note that the total energy does not include the energy from interaction with solvent. I will discuss how the solvent gets involved in the energy calculation of molecular systems in Chapter II.

The energy functions discussed above are not of any value if they are not accompanied with a set of parameters, such as k_θ and θ_0 that describe the energetic properties of the interacting particles. The aim of optimisation of forcefield parameters is to adjust the parameter values so the forcefield is able to match up experimental data. This may include

the use of experimental spectroscopic, thermodynamic, and crystallographic data as well as data computed using quantum mechanics methods. The AMBER, CHARMM, OPLS-AA and GROMOS forcefields are each based on a different type of experimental data, although there is some overlap. The parameters for these forcefields were extensively optimised with particular emphasis on the treatment of proteins.

3. Molecular dynamics(MD) and Langevin dynamics(LD)

The molecular dynamics simulation technique is one of the most important and widely used methods in studying the biological systems at atomic detail.

The molecular dynamics simulation method is based on Newton's second law of motion. After one sets initial positions, velocities as well as forcefield I just described, he or she is capable of obtaining a trajectory that describes the positions, velocities and accelerations of the particles as a function of time. It is the most accurate way to describe the time evolution of a system of chemically reacting molecules. The fundamental equations are following

$$\mathbf{F}_i = -\nabla_i U \tag{I.4}$$

and

$$\frac{\mathbf{F}_i}{m_i} = \frac{d^2\mathbf{r_i}}{dt^2} \tag{I.5}$$

where force $\mathbf{F}_i$ is the force acting on the ith particle in the system is derived from the potential energy $U(r^N)$ in Eq. (I.4) which is defined by the selected forcefield, where $r^N = (r_1, r_2, r_3 \cdots r_N)$ represents the complete set of $3N$ atomic coordinates. The m_i is the mass for the ith particle. Even though the equation is simplistic in its form (Newton's law), there is no analytical solution to the equation of motion for systems of more than two particles, so it must be solved numerically. Numerous algorithms have been developed for solving these equations, and most are derived from the widely used Verlet algorithm [18], which requires the knowledge of current positions, $\mathbf{r}(t)$; acceleration $\mathbf{a}(t)$; and the position from the previous step, $\mathbf{r}(t-\delta t)$ .The position of the next step can then be found using the formula,

$$\mathbf{r}(t+\delta t) = 2\mathbf{r}(t) - \mathbf{r}(t-\delta t) + \delta \mathrm{t}^2 \mathbf{a}(t) \tag{I.6}$$

where **a**(t) is calculated from Eq. (I.5). The velocities can be calculated by the difference in position at time $t + \delta t$ and $t - \delta t$

$$\mathbf{v}(t) = \frac{\mathbf{r}(t + \delta t) - \mathbf{r}(t - \delta t)}{2\delta t} \tag{I.7}$$

Several variations on the Verlet algorithm have been developed. These algorithms include leap-frog algorithm [19], velocity Verlet algorithm [20] and Beeman's Algorithm [21], which are not discussed in here.

All above methods are based on Newtonian mechanics, in which the motion of the system of interest is fully deterministic given initial conditions. But interactions with the environment will affect any system. In reality, people mostly investigate solute-solvent systems in which the primary focus is the behaviour of the solute. But the solvent influences the motion of solute by collisions. The collisions cause two major effects. One is the frictional drag force which slow down the solutes. The higher the velocity, the larger the restoring force, which is expressed as a frictional force. If particles moving through a fluid at relatively slow speeds where there is no turbulence, the force of drag is approximately proportional to velocity, but opposite in direction. The equation for viscous resistance is

$$\mathbf{F}_{\text{friction}} = -\alpha \mathbf{v} \tag{I.8}$$

where **v** is the velocity and α is the friction coefficient. This friction coefficient could also be written as

$$\alpha = m\gamma \tag{I.9}$$

where m is the mass of the particle and γ^{-1} turns out to be proportional to the time taken for the particle to lose memory of its initial velocity (the velocity relaxation time). The larger the viscosity the larger is γ. From the macroscopic Stokes law the friction coefficient is defined as

$$\alpha = 6\pi\eta a \tag{I.10}$$

where a is the particle radius and η is viscosity.

Another effect comes from random forces R_i(t) on solutes which will decrease with lower temperature. The Langevin equation, which describes the stochastic dynamics, takes care of these two effects. The Langevin equation is described as below [22]

$$m_i \frac{d^2\mathbf{r}_i}{dt^2} = \mathbf{F}_i - m_i\gamma_i \frac{d\mathbf{r}_i}{dt} + \mathbf{R}_i(t) \tag{I.11}$$

The $\mathbf{F}_i$ includes both external forces and interactions between the particle and the other particles. The latter depends on the position of the particle and is estimated by $\nabla U(\mathbf{r})$, which is the particle interaction potential. Both are deterministic, thus this term comes from Newtonian mechanics. There are two additional terms due to the solvent. The frictional term is proportional to velocity, and the random force $\mathbf{R}$ is often assumed to be uncorrelated with the particle velocities, positions and the forces acting on them. It is modeled by stochastic process as satisfying

$$\langle \mathbf{R_i}(t) \rangle = 0 \tag{I.12}$$

where $\langle\rangle$ means an ensemble average of force realizations which are detected by using a collection of particles at time t. In other words, the random force could be any distribution as long as it has zero mean. Successive impacts are assumed uncorrelated

$$\langle R(t)R(t') \rangle = \Gamma\delta(t - t') \tag{I.13}$$

where Γ describes the strength of the random force. It is worth noting that the friction term and random force term are related

$$\Gamma = 2\gamma m k_B T \tag{I.14}$$

Such a relation connecting the strength of fluctuations to the systematic dissipation (viscosity) was derived by Einstein. Each particle gains energy from the bath (random forces) and releases this energy back to the bath due to dissipative (viscosity) forces. The relation between the strength of the random forces and viscous forces, with a dependence on temperature manifests the fluctuation-dissipation theorem.

4. Time correlation

It is obvious that the configurations generated from simulation are correlated in time. The waiting time for the system to fully forget previous steps is called "correlation time". Since one of the important parts of my work is to analyze sampling efficiency, good understanding of auto-correlation function is very necessary.

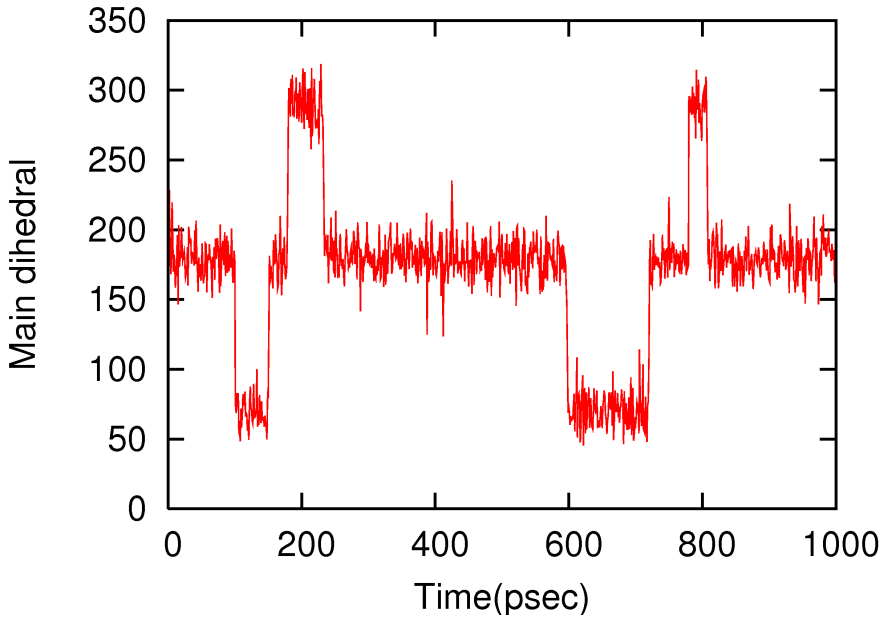

Figure 2: The main dihedral of butane, which is the most important coordinate. The correlation time for this system is approximately 150∼200 psec.

The Fig. 2 shows a time series of main dihedral coordinate in a Langevin simulation. Butane is a very simple system and it has three different states defined only by its main dihedral angle. The Fig. 2 shows that it needs some time (steps) before moves to other states. They could be correlated even in different states. This fact indicates that there are correlations in time among the configurations. In other words, it will take some time for the system to forget previous behaviour. Quantitative measurement of correlation time has been explored. Let us first define the difference between the two times $\tau = t_2 - t_1$, which influences the degree of correlation. We need to examine all pairs of time, $f(t)$ and $f(t+\tau)$, over all possible t values, where $f = f(\mathbf{r}^N)$. So the correlation function is defined as

$$C_f(\tau) \equiv \frac{\langle f(t)f(t+\tau)\rangle - \langle f(t)\rangle\langle f(t+\tau)\rangle}{\langle f(t)f(t)\rangle - \langle f(t)\rangle\langle f(t)\rangle} \tag{I.15}$$

$$= \frac{\langle f(t)f(t+\tau)\rangle - \langle f\rangle^2}{\langle f^2\rangle - \langle f\rangle^2} \tag{I.16}$$

where the bracket "$\langle\rangle$" means that the averages are to be performed over all t values with fixed time interval τ. This correlation function is defined under the assumption that the correlation depends only on the time-distance between the pair of values but not on their position in time, *i.e.,* it has to be in a stationary state. Note that it is auto-correlation function since it measures the self-correlation. The $C_f(0) = 1$ means it is fully correlated with itself, while it will decrease as τ increases and lower limit is 0 when τ goes to infinity. The correlation time t_{corr} (the time for the molecule to "lose" memory) is quantitatively defined by

$$t_{\text{corr}} = \int_0^{+\infty} d\tau C_f(\tau) \tag{I.17}$$

where heavy calculation or numerical integration will often be required. In most cases, the correlation function can be fit to a presumed functional form, such as an exponential. Thus a rough estimate of correlation time is the τ value for which C is e^{-1}. There are some other approaches proposed about the correlation time. Block averaging method was proposed by Flyvbjerg [23], which emphasizes the correlated data analysis. Hess applied principle component analysis (PCA) to find the correlation time [24]. Recently Lyman and Zuckerman developed structural histograms methods [25, 26] to estimate the correlation time and thus the effective sample size. In this method, they divided the whole configuration space into several bins and bin populations were arrayed as a one dimensional structural histogram reflecting the full configuration-space distribution. They then applied binomial and related statistics to estimate correlation time from bin population variances. The advantage of this method is that it is applicable to quite complex systems without invoking correlation functions, which need a significant amount of data and calculation.

There is a related point. Different people will study different correlation functions/times when calculating different properties of a system. But the correlation time varies a lot among different coordinates since they have different correlation functions. For example, a rapidly

oscillating bond will converge very fast, while the dihedral angle, which is highly coupled with other coordinates, needs much longer time to converge. Note that the convergence I am discussing is not the absolute convergence, rather "relative convergence". While a simulation reaches convergence, it starts to correctly generate canonical sampling in Boltzmann distribution, and the simulation process is in the true sampling regime – it is long enough to produce multiple properly distributed statistically independent configurations. It will take even a lot longer time for the whole system to converge. I will discuss in detail in Chapter IV.

5. Limitation of Molecular and Langevin Dynamics simulations

There has been debate about the importance of simulation in molecular biophysics. In fact, there is some limitation of such simulations even though they are extremely useful and widely accepted in the fields of physics, chemistry and engineering.

The power of current CPU limits the MD simulation of typical biomolecules to the micro-sec scale. The standard time length for each MD step is one femtosecond, which is the waiting time for the bond-length to equilibrate. The proteins function in 1 second to 100 seconds scale,which are 10^{14} to 10^{17} steps [27]. It is several orders higher than we can generate. This largely limits us from investigating large biological systems at the atomic level and compare with outcome from experiments. From another prospective, the number of local energy minima increase exponentially as the number of atoms of the protein increase. A system with a large number of low barriers would need substantial time to sample the whole configuration space [28]. It is also possible that a trajectory which appears long enough (for instance 1 μs) is proven inadequate with additional simulation [29]. All the above reasons contribute to the sampling problem, which will be discussed in detail in Chapter II.

Even with these disadvantages, the simulation is not "useless". However, it is very crucial in determining protein structures, and offer valuable insights into fast local motions. The bright side is that more and more algorithms have been improved to simulate longer trajectories, which will be discussed in Chapter II.

6. Algorithms beyond MD and LD

Both MD and LD are dynamic simulations, where the configurations in the ensemble are correlated in time. People are developing many new algorithms every day. Here I briefly introduce two other important algorithms.

The first one is "Replica Exchange" (parallel tempering). It was described by Sugita, Swendsen and their coworkers [30, 31] and is a powerful technique able to enhance conformational sampling, compared with parallel simulations at different constant temperatures. In this method, several independent replicas of the system are simulated in parallel, where each replica evolved at a different temperature. At selected times a swap is performed between the replicas, and this exchange is accepted or rejected based on a Metropolis acceptance criterion [32]. This method has been widely used in atomistic level simulations and even applied in coarse-grained models to study protein folding. Since configurations could be produced at different temperatures, non-sequential correlation will be brought in. Thus replica exchange is considered as a non-dynamical method.

Another method is "library based Monte Carlo" (LBMC) simulation, developed by Zuckerman's group [33]. It performs Boltzmann sampling of molecular systems based on precalculated statistical libraries of molecular-fragment configurations, energies, and interactions. A protein is divided into several fragments which are pre-sampled, canonically and independently. The Boltzmann-distributed library for each fragment accounts for all correlations internal to the fragment. Trial moves consists of swapping the present configuration of one or more fragments with elements in the corresponding libraries. Then a Metropolis-Hastings criterion [34] for an LBMC trial move is used in order to meet detailed-balance conditions. After many trial moves, the configurations collected from the accepted moves consist of the desired Boltzmann-distributed ensemble. LBMC can be applied to both atomistic and coarse-grained models for flexible peptides [35], and it is much faster than standard Monte Carlo simulations.

In my Ph.D study, I was also involved intensively in development of a growth algorithm, which extends the polymer-growth algorithm [34, 36–48] into biological system sampling. It will be discussed in Chapter II.

C. SAMPLING EFFICIENCY ESTIMATION

The above section gives a brief introduction on the theory of dynamic simulations and Langevin dynamics, and there are a lot of non-dynamic algorithms as well, such as the growth algorithms that I will discuss next chapter. One thing one is interested is to find out whether these "fancy" algorithms are better than standard MD or LD or which algorithm is "faster" than others. Since almost all the algorithms will give correlated final configurations, the actual number of independent configurations is less than the number of final configurations. Thus one does not know the efficiency unless he or she can calculate effective sample size (ESS), which is defined as the number of independent configurations that an ensemble is roughly equivalent to. After ESS is calculated, one can estimate efficiency among different algorithms by

$$\text{Cost per config} = \frac{\text{Total cost of simulation}}{\text{ESS}} \tag{I.18}$$

where "cost per config" is time cost to get one independent configuration. Calculation of ESS is the key of sampling efficiency estimation. In a dynamical simulation, a conventional view of sample size is given by the following equation,

$$\text{ESS} = \frac{t_{\text{sim}}}{t_{\text{corr}}} \tag{I.19}$$

where t_{sim} is the simulation time, and t_{corr} is the correlation time of the variable measured in the system. Thus, significant effort has been invested in developing methods to calculate the correlation time as I mentioned before. I also mentioned in previous section that different variables (coordinates) have different correlation times. The t_{corr} in Eq. (I.19) should be the largest correlation time among all the variables. In other words, the whole ensemble should be converged in order to estimate the largest correlation time. In practice, it is very hard due to strongly coupled coordinates. Furthermore, this approach applies to dynamic simulation. Thus a standard and universal method for both dynamic and non-dynamic simulations is needed. I have been working extensively on the effective sample size calculation throughout my Ph.D study. The details will be provide in Chapter IV and V.

D. OUTLINE OF THESIS

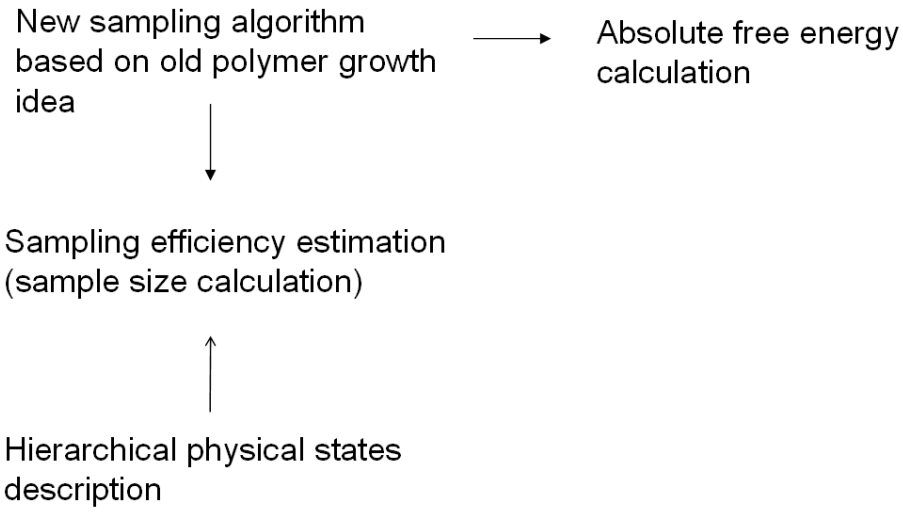

Figure 3: The relation among projects that have been accomplished

The outline of this thesis and relation among the projects are shown schematically in Fig. 3. It is organized as follows. In Chapter II ("New sampling algorithm based on old polymer growth idea" in Fig. 3), I will review polymer-growth approach and introduce the novel sampling algorithm, which incorporates polymer growth idea into biological system sampling. In Chapter III ("Absolute free energy calculation" in Fig. 3), I will discuss the methods and implication absolute free energy calculation using the sampling algorithm. Sampling efficiency will be reviewed and studied in chapter IV. Furthermore, I will discuss a universal effective sample size calculation method. A related subject, discovery of "physical states" will be explored in Chapter V. I will focus on the principles and methods of physical states discovery in this chapter. Summary and future projects are included in Chapter VI.

II. NOVEL SAMPLING OF ALL-ATOM PEPTIDES USING A LIBRARY-BASED POLYMER-GROWTH APPROACH

A. OVERVIEW

This chapter investigates whether decades-old polymer-growth algorithms [34, 36–48] have promise for the study of biomolecules modeled by modern atomistic forcefields. Although polymer approaches have previously been applied to peptides [49–51], their application to atomistic forcefields at physiological temperatures has been problematic [52–54]. Here I report a novel implementation of growth algorithm based on pre-calculated statistical libraries of molecular fragment configurations and energies. The encouraging results from a limited set of small test peptides, reported below, suggest that further investigation is warranted. This chapter has been adapted from a published paper [55].

1. Historical background of Polymer growth

The well-known problem of sampling biomolecules typically has been addressed by dynamical simulations and variants – molecular dynamics (MD), Langevin dynamics (LD), and Metropolis Monte Carlo with local moves. All these approaches suffer from the well-known problem of undersampling: dynamical simulations of proteins are far too short to probe timescales (and motions) thought to be of dominant biological importance. Even simulations of modest-sized peptides are slow to "converge" [25, 26]. Sophisticated variants of dynamical simulations, such as replica exchange [30, 56–58], also have not convincingly solved the undersampling problem [59–61]. While multi-resolution methods appear to have substantial promise [59, 62–64], rigorous applications have been restricted to small systems thus far.

The importance of sampling biomolecules and the intrinsic limitations of dynamical simulation together suggest the value of exploring fully non-dynamical polymer growth algorithms. Such methods have a history dating back more than fifty years. Initial studies focused on straightforward build-up of lattice-polymer chains [36, 37, 39], but the early approaches were limited by the "attrition problem" in which the vast majority of chains encounter dead ends before reaching a significant size. Our own approach builds directly on methods developed to treat attrition, especially (i) the Rosenbluths' approach of re-weighting chains based on possible growth steps [40], and (ii) equally seminal work by Wall and Erpenbeck describing "enrichment" of successful partially grown chains by replication and appropriate weighting [42]. Wall, Rubin and Isaacson noted that future increments of the growth of a lattice polymer were limited to a small set of configurations [41], partly anticipating the libraries we employ here. Many additional improvements have also been proposed [43–45]. The basic theory behind polymer growth as we apply it, along with key practical insights, was fully set out by Garel and Orland in 1990 [46]. Important descriptions of growth algorithms are also provided by Grassberger [47] and by Liu [34].

2. Application of Polymer growth in biological system sampling

Polymer growth algorithms have been applied previously to biomolecules. Highly simplified models of proteins were studied by Grassberger and coworkers [49] and by Liu and coworkers [50, 51, 65]. Garel, Orland, and coworkers applied polymer growth methods to all-atom peptide models – but their work employed extremely high-temperature sampling ($T = 1000K$) followed by energy minimization [46, 52–54]. The use of pre-calculated fragment libraries emulates ideas from the ROSETTA software [66] as well as from work by Clementi and coworkers [67, 68]. However, none of these previous studies appears to have generated canonical sampling for a modern atomistic forcefield at $T \sim 300K$.

In light of the significant body of historical work, the present contribution must be considered pragmatic rather than theoretical. In brief, this reported work shows that pre-generated libraries of statistically distributed monomer fragment configurations can be used in implicit solvent sampling of all-atom molecular systems at temperatures of interest ($T = 300K$). For

high quality statistical sampling the present implementation is limited to small peptides – up to about eight residues and less than 100 atoms. However, besides equilibrium sampling, the growth procedure can be also used for rapid generation of approximate (i.e., steric-clash free) ensembles of larger peptides containing up to 16 amino acid residues. Although the present work is formally similar to Zuckerman and coworkers' previous use of fragments for free energy calculations [69], this study presents critical technique improvements which greatly improve efficiency.

This present study also employs recently developed statistical approaches to quantify the degree to which efficiency has been gained. The library-based strategy is shown to be extremely efficient in some cases – decreasing the required wallclock time by over one order of magnitude. However, I believe that several improvements are possible, as described in the Sec. II.E. In this approach the choice of fragments is flexible and they can correspond to different groups of atoms in the molecule. For proteins the natural choice of fragments is the amino acid residues because proteins consist of only 20 building blocks. However, other choices are possible. When the fragments correspond to the backbone and side chains, the procedure is essentially a multi-resolution method. The backbone can be sampled using other methods such as Zuckerman's previously developed library-based Monte Carlo [33], followed by the gradual addition of more atomistic detail embodied in side chains.

3. Work I contributed in this project

This work is not done all by myself. My contribution to this project is to develop this algorithm and apply it to small systems. I combined two butane(C_4H_{10}) ensembles into an octane(C_8H_{18}) ensemble and further combined two octane ensembles into a cetane ($C_{16}H_{34}$). The results are not published since people are more interested in peptide systems, but they will be presented later in Sec. II.D. Dr. Artem Mamonov, a postdoc in Zuckerman group, took over my work and applied this algorithm to bigger and more complex peptide systems. We both work on the improvement of algorithms such as "dummy" atoms and "optimal resampling". I will show the details of this method below.

B. FORMALISM

As noted in the Sec. II.A, polymer growth algorithms have been developed and used over decades [34, 36–48]. The present approach follows earlier work in many regards, but is specifically tailored to the use of modern atomistic forcefields and implicit solvent. The presentation of the algorithms relies solely on straightforward re-weighting concepts [34, 70]. I describe a simple and apparently novel approach to using libraries of molecular fragments which can save significant computational cost.

1. Forcefield,fragments and notation

In this study, I generate equilibrium configurations according to the OPLS-AA forcefield [15] using a simple implicit solvent model (with uniform dielectric constant of 60) at 298 K. This dielectric constant has been chosen to give reasonable agreement for Ramachandran propensities as compared to GBSA solvent model [71].

The potential energy of the forcefield plus the solvent model will be denoted by $U(\mathbf{x})$,where the full set of $3N-6$ internal coordinates $\mathbf{x} = (x_1, x_2, \cdots, x_{3N-6})$, consists of $N-1$ bond, $N-2$ bond angles and $N-3$ dihedrals. The full set of coordinates corresponding to a single molecular fragment y will be denoted by $\mathbf{x}_y$ with $y = A, B, C, \cdots$. The collection of forcefield terms for fragment y, denoted by U_y will contain all terms internal to the particular subset of atoms included in the fragment. That is, it will include all bonded and non-bonded terms for those atoms. Dummy atoms may be added to a fragment, as in the present study, to include the six degrees of freedom that specify the orientation of fragments relative to each other. However, dummy atoms will have no effect on the trial distribution.

There is an assumption that fragments are non-overlapping and exactly divide all coordinates, so that for the whole molecule the full set of coordinates may be written as

$$\mathbf{x} = \mathbf{x_A}, \mathbf{x_B}, \mathbf{x_C} \tag{II.1}$$

It is important to realize that the full forcefield U can never be written as a sum of fragment forcefields . The reason is that, regardless of which intermediate coordinates are included via

dummy atoms, no coordinate set includes distances between atoms from different fragments. Needless to say, such inter-atomic distances are fundamental to the full molecular forcefield. Inter-fragment interactions are fully accounted for in the growth procedure, as described below.

2. Combination of fragments

In this approach, a molecule is sampled by growing it from scratch using pre-calculated molecular fragments. Here we describe the process of joining fragments which may be repeated inductively by adding additional "monomers" onto the growing chain. Configurations for each fragment are calculated in advance so that they are distributed according to the Boltzmann factor of the forcefield describing the isolated fragments. The set of Boltzmann-distributed configurations for each fragment is called a "library".

The basic procedure for joining fragments is simple. A new fragment configuration is drawn with uniform probability from its library and added to the partially grown chain ensemble. The interaction energy between the new fragment and other previously added fragments is evaluated. The generated configurations are reweighted to the Boltzmann factor distribution describing the partially grown molecule to correct for the new interactions.

Consistent with free energy calculations using the growth process [69], we will define a set of intermediate models U_j which correspond to different stages of the growth process. We note that these intermediates are a little different than employed (before) in ref [69].

For a molecule consisting of k fragments, we will employ k intermediate models with interactions between fragments gradually "turned on". The first intermediate , sampled at the library generation stage, includes interactions internal to each fragment, while subsequent intermediates add the indicated interactions among fragments $A, B, C, \cdots$. These

intermediate models can be written as

$$\begin{aligned} U_1(\mathbf{x}) &= U_A(\mathbf{x}_A) + U_B(\mathbf{x}_A) + U_C(\mathbf{x}_C) + \cdots \\ U_2(\mathbf{x}) &= U_1(\mathbf{x}) + U_{AB}(\mathbf{x}_A, \mathbf{x}_B) \\ U_3(\mathbf{x}) &= U_2(\mathbf{x}) + U_{AC}(\mathbf{x}_A, \mathbf{x}_C) + U_{BC}(\mathbf{x}_B, \mathbf{x}_C), \\ &\cdots \\ U(\mathbf{x}) &= U_{k-1}(\mathbf{x}) + \sum_{y=A,B,\cdots} U_{yz}(\mathbf{x}_{yz}) \end{aligned} \tag{II.2}$$

where U_{yz} denotes all forcefield interaction terms between fragments y and z. The last intermediate $U_{\mathbf{x}}$ is simply the full molecule and the sum $\sum_{y=A,B,\cdots} U_{yz}(\mathbf{x}_{yz})$ represent interactions between the last fragment z and all other fragments in the molecule.

3. Growth by reweighting

The polymer-growth approach heavily relies on the re-weighting concept [34, 70] because interactions between fragments are not included in the libraries of individual fragments. In essence we generate configurations with non-interacting fragments and gradually reweight them into an ensemble with all interactions. In other words the purpose of reweighting is to effectively put back all the interactions and correlations between fragments into the molecule.

At each stage, we want to generate a suitably distributed ensemble – called the target ensemble $P_j^{\text{targ}} \propto \exp[-\beta U_j(\mathbf{x})]$ for stage j with the set U_j defined in Eq.(II.2). When $j < k$, this target ensemble based on U_j includes interactions only for the partially "grown" molecule. Yet configurations for stage j, as will be seen, are generated according to a different distribution, denoted P_j^{gen}. Hence, configurations must be reweighted according to

$$u_j(\mathbf{x}) = \frac{P_j^{\text{targ}}(\mathbf{x})}{P_j^{\text{gen}}(\mathbf{x})} \tag{II.3}$$

where $u_j(\mathbf{x})$ is the weight of a configuration at stage j. (In fact, as explained below, $u_j(\mathbf{x})$ is an intermediate weight.) In Eq (II.3) and subsequent equations, the symbol $\mathbf{x}$ does indeed represent the full set of coordinates. In intermediate stages $j < k$, however, some interactions are omitted: see Eq.(II.2).

To perform the reweighing procedure, one needs to define the $P^{\text{gen}}$and P^{targ} for each intermediate stage. Let us consider each stage in detail. The first stage U_1 includes interactions within each fragment which are sampled at the library generation stage. The second stage U_2 corresponds to turning on interactions between fragments A and B, starting from configurations already distributed according to U_1. Thus the generating distribution P_2^{gen} is simply proportional to the Boltzmann factor describing the first intermediate with non-interacting fragments:

$$P_2^{\text{gen}}(\mathbf{x}) \propto \exp(-\beta U_1(\mathbf{x})). \tag{II.4}$$

The distribution targeted at the second stage P_2^{targ} proportional to the Boltzmann factor describing the second intermediate:

$$P_2^{\text{targ}}(\mathbf{x}) \propto \exp(-\beta U_2(\mathbf{x})). \tag{II.5}$$

At the third stage, similarly, interactions are turned on between fragment C and previously combined fragments A and B. As before P_3^{gen} is nothing but P_2^{targ}

$$P_3^{\text{gen}}(\mathbf{x}) = P_2^{\text{targ}}(\mathbf{x}) \propto \exp(-\beta U_2(\mathbf{x})) \tag{II.6}$$

Likewise, P_3^{targ} distribution is proportional to the Boltzmann factor describing the third intermediate:

$$P_3^{\text{targ}} \propto \exp(-\beta U_3(\mathbf{x})). \tag{II.7}$$

It is not difficult to generalize this combination process for any other intermediate. For the kth intermediate (corresponding to the full molecule) P_j^{gen} and P_j^{targ} can be written as

$$P_k^{\text{gen}}(\mathbf{x}) = P_{k-1}^{\text{targ}} \propto \exp(-\beta U_{k-1}(\mathbf{x})) \tag{II.8}$$

$$P_k^{\text{targ}}(\mathbf{x}) \propto \exp(-\beta U(\mathbf{x})). \tag{II.9}$$

It is important to note that in this procedure P^{gen} is built sequentially based on P^{targ} from the previous stages. This is the essence of "sequential importance sampling" i.e., the probability distribution of the full molecule is built sequentially step by step. The advantage

of sequential importance sampling is that the probability distribution is changed in small increments to give the better overlap between P^{gen} and P^{targ} at each stage.

The required partial weights u_j can be calculated based on the incremental weights of Eq. (II.3). Specifically, the weight of a configuration at stage j can be written recursively based on the weights from previous stages:

$$w_j = w_{j-1} u_j. \tag{II.10}$$

Substituting the corresponding P^{gen} and P^{targ} from Eqs. (II.4) – (II.9) into Eq. (II.10) the partial weights can be written as

$$\begin{aligned}
w_1(\mathbf{x}) &= 1 \\
w_2(\mathbf{x}) \propto w_1(\mathbf{x}) \frac{\exp(-\beta U_2(\mathbf{x}))}{\exp(-\beta U_1(\mathbf{x}))} &= w_1(\mathbf{x}) \exp(-\beta U_{AB}(\mathbf{x_A}, \mathbf{x_B})) \\
w_3(\mathbf{x}) \propto w_2(\mathbf{x}) \frac{\exp(-\beta U_3(\mathbf{x}))}{\exp(-\beta U_2(\mathbf{x}))} &= w_2(\mathbf{x}) \exp[-\beta(U_{AC}(\mathbf{x_A}, \mathbf{x_C}) + U_{BC}(\mathbf{x_B}, \mathbf{x_C}))], \\
&\cdots \\
w(\mathbf{x}) \propto w_{k-1}(\mathbf{x}) \frac{\exp(-\beta U(\mathbf{x}))}{\exp(-\beta U_{k-1}(\mathbf{x_{k-1}}))} &= w_{k-1}(\mathbf{x}) \exp[-\beta \sum_{y=A,B,\cdots} U_{yz}(\mathbf{x}_y, \mathbf{x}_z)]
\end{aligned} \tag{II.11}$$

where $w(\mathbf{x})$ is the total weight for the full molecule i.e., with interactions "turned on" between all fragments. Note that $w_1(\mathbf{x})$ is equal to one by construction because fragment configurations in the libraries are distributed according to the corresponding P^{targ}– i.e., the Boltzmann factor describing the individual fragments.

The "resampling" protocol, described later, will use the partial weights w_j. However, it is instructive to note that the total weight $w(\mathbf{x})$ in Eq. (II.11) can be re-written by expanding the weights and rearranging terms, resulting in

$$w(\mathbf{x}) \propto \frac{\exp(-\beta U(\mathbf{x}))}{\exp(-\beta U_1(\mathbf{x}))} \tag{II.12}$$

Eq. (II.12) shows that the total weight takes into account all the interactions missing in the non–interacting fragments described by the first intermediate U_1.

Note that the weights in Eqs. (II.11) and (II.12) are proportional to the ratio of the Boltzmann factors up to the constant, which is the ratio of the corresponding partition functions. However, this constant is not needed for re-weighting because only the relative weights are important.

4. Resampling

In general, configurations with low weights have low importance in the ensemble and therefore it is desirable to save computer time by eliminating such configurations from future consideration. However, such elimination must be performed statistically to preserve the correct distribution [34]. Such a "resampling" process refers to eliminating, duplicating, and/or adjusting weights of configurations in the original ensemble resulting into an alternative ensemble [34]. Both ensembles are formally equivalent in representing the desired distribution.

A number of resampling algorithms have been suggested in statistics and data processing [34, 72]. We implemented several resampling schemes in the growth algorithm and found a scheme termed "optimal resampling" [72] to be the most efficient. The advantage of optimal resampling compared to other schemes is that it guarantees distinct configurations and at the same time allows a large diversity of weights.

The main feature of optimal resampling is that it guarantees drawing the desired number of distinct configurations, denoted by M, from an original ensemble containing N configurations and corresponding weights. This is achieved by employing a threshold weight c which uniquely defines M. The configurations are accepted with probability $\min\{1, \frac{w_j(\mathbf{x})}{c}\}$, where $w_j(\mathbf{x})$ are the partial weights at stage j. The resampling of only distinct configurations is guaranteed by employing a special numerical cumulative distribution function (cdf) [72].

We implemented the optimal resampling in the growth algorithm at the end of each combination stage. After the fragments are joined and the weights are calculated, the configurations are resampled into a smaller ensemble containing 10 percent of the original configurations. The 10-fold reduction factor was found to be the most efficient based on trials of different N and M values. The typical ensemble size employed in the simulations is $N = 10^5$ configurations, which is resampled into an ensemble of size $M = 10^4$. As described later, an "enrichment" procedure is employed to compensate for configurations eliminated by resampling and to maintain a constant ensemble size at different combination stages.

It is worth noting that after the last combination stage, configurations with weights may be resampled into an ensemble without weights. We implemented several different resampling

algorithms to eliminate weights in the final ensemble. However, we consistently found that such resampling considerably reduces information contained in the weights. Therefore, after the last combination stage we use the same optimal resampling scheme as at other stages and save configurations with weights for further analysis. This is similar to keeping a larger number of correlated "snapshots" from a dynamics trajectory [73].

5. Approximate ensemble

Besides equilibrium sampling, the growth procedure can be adapted for rapid generation of approximate ensembles. This may be useful for larger systems for which precise ensembles are not required – for instance, in schemes which assemble protein configurations from multi-residue segments [66, 74–76]. The only new feature of the approximate procedure is that after the last combination stage, configurations are used without weights. This way weights are used only to identify configurations without steric clashes. In other words, resampling works as a "bump check".

6. Assessment of sampling precision and efficiency

In the present work efficiency of the growth algorithm is defined as the savings in wallclock time to achieve the same level of statistical precision in sampling of configuration space distribution relative to standard Langevin dynamics. This precision can be quantified by the number of statistically independent configurations contained in the trajectory (i.e., effective sample size (ESS)). To assess efficiency, time to generate a single statistically independent configuration can be compared between two methods. Thus, efficiency is defined as

$$\gamma = \frac{t_{Langevin}}{t_{Growth}} \frac{ESS_{Growth}}{ESS_{Langevin}} \tag{II.13}$$

where $ESS_{Langevin}$ and ESS_{Growth} are the effective sample sizes of the growth and Langevin simulations respectively. The symbols t_{growth} and $t_{Langevin}$ denote wallclock times of growth and Langevin simulations respectively.

To calculate the ESS for both growth and Langevin simulations I used a recently developed statistical analysis. Qualitatively, the idea is to divide configuration space into

approximate physical states and calculate variance in each state. The variance is inversely proportional to the effective sample size. The approximate physical states can be constructed using Voronoi bins in configuration space. The reference structures for the Voronoi procedure [77] are derived from the protocol described in Ref. [26].

To check the results of the previous method we also used a second method to calculate the ESS for Langevin simulations. This method employs the previously developed "de-correlation" time analysis and can be used only for dynamic simulations [26]. Briefly, the idea is to determine how much simulation time must elapse between configurations in the trajectory in order for them to exhibit the statistics of fully independent samples. Using the de-correlation time and the total simulation length the number of statistically independent configurations in the trajectory can be calculated.

Effective sample size calculation will be discussed in detail in Chapter IV.

C. IMPLEMENTATION

The growth formalism described in Sec. II.B does not lead to a unique algorithm, but can be implemented in many different ways. Implementation details are particularly important because modern forcefields are much more complicated than the early simple polymer models. Indeed, in the study we found that the efficiency of the growth algorithms depends significantly on the implementation. Here, we describe the technical approaches that helped to significantly improve the efficiency of the growth algorithm.

1. Fragments libraries

The advantage of using libraries is that some interactions and, therefore correlations within a molecule, can be calculated in advance and then used in multiple simulations saving CPU time. Instead of generating new fragment configurations on the fly, they can be cheaply retrieved from the memory. This approach is well suited for proteins, which consist of only 20 different building blocks. We can build up libraries for different amino acids and then

combine them according to the sequence to sample any peptide or protein. The idea to use molecular fragments in molecular simulations is well established in the literature [78, 79] and has been successfully implemented in the protein structure prediction software Rosetta [66]. Earlier we have used libraries in a Monte Carlo approach [33].

Fragment libraries can be generated using any canonical method such as Langevin dynamics or Metropolis Monte Carlo. The only requirement for the libraries is that they should represent the true equilibrium distributions. In practice we used internal coordinate MC because it allows fixing some degrees of freedom such as some bond angles and dihedrals introduced with the dummy atoms. The dummy atoms were employed for two reasons. First they provide the six degree of freedom that specify the orientation of fragments relative to each other. Second, the dummy atoms were chosen to interact with the real fragment atoms to provide better overlap with the full molecule distributions. We used libraries containing 10^5 configurations.

We note that the fragments contain the same degrees of freedom and are sampled according to the same forcefield as employed in previous study [69]. The only difference is that in previous work the fragment libraries were generated by sampling the internal coordinates independently with subsequent reweighting into the full fragment distributions.

2. Enrichment

Enrichment entails making multiple copies of configurations at different stages of growth without introducing statistical bias, in order to increase the chances of partially grown chains to survive [80]. We implemented enrichment in the growth algorithm and found that it significantly increased the efficiency. One drawback of enrichment is that when chains are replicated, they are no longer statistically independent, limiting how much enrichment can ameliorate attrition. If chains are replicated too much, the configurations become too statistically correlated, and ultimately limit efficiency. We found that a very efficient implementation of enrichment in this growth algorithm is when it is applied after each combination stage and chains are replicated 10–100 times.

3. Recycling the energy terms

In addition to coordinates, the potential energy of each fragment configuration can be calculated in advance and stored in the libraries. When fragments are combined, the potential energy of each fragment configuration can be cheaply retrieved from the computer memory saving CPU time. However, these savings will only be moderate for long molecules containing many fragments because interactions between fragments will dominate. We implemented recycling of energy terms in the growth algorithm and found that it helped to increase the efficiency for all the systems studied.

4. Cartesian and internal coordinates

To implement the growth formalism of Sec. II.B, it could seem natural to use internal coordinates, particularly for connecting fragments. However, each configuration ultimately must be converted to Cartesian coordinates for potential energy evaluation. In the original implementation fragment configurations were combined in internal coordinates and then converted to Cartesian for energy calculation. But we found that a large fraction of CPU time was actually spent on coordinate conversion.

The efficiency of the growth procedure was significantly improved when fragments were combined in Cartesian coordinates. This was implemented by storing "connector coordinates" – *i.e.,* the six relative degrees of freedom – along with transformation matrices for each fragment configuration. First, the six degrees of freedom that specify the orientation of fragments relative to each other were used to set up the local coordinate systems. Given the local coordinate systems for each fragment, the appropriate transformation matrices were applied to generate the full Cartesian coordinates. In practice, configurations in the libraries were pre-oriented in the local coordinate system at the N-terminus of the residue based fragments and only one transformation matrix (at the C-terminus) was saved for each configuration in the library.

All transformation matrices were calculated using quaternion operations which allow fast and accurate transformations [81].

5. Software optimizations

The cost analysis of the growth algorithm revealed that it is "memory bound" – *i.e.,* the bottleneck is not the CPU operations but rather the transfer of data from memory to CPU. It is memory bound because it heavily relies on pre-calculating and storing configurations and energies in the memory. The transfer rate of data between the main memory and CPU is limited and becomes the bottleneck. To hide the memory latency problem modern CPUs utilize "cache" memory which allows much faster communication with CPU. However, the size of cache is much smaller than the main memory size so the data can be cached only in relatively small chunks. The memory bound algorithms can be improved by reusing the data and "neighbor use". Reuse helps to reduce the transfer of data from main memory to CPU by reusing as much as possible the data stored in cache and CPU registers. Neighbor use helps to perform computation on data (physically) close in memory reducing the transfer of data from memory to cache.

We implemented several standard optimization techniques in the C code including array linearization and blocking [82] both aimed at improving the reuse and neighbor use of fragment configurations and energies stored in the libraries.

6. Breadth and depth

The growth algorithm can be implemented in two different ways: "breadth first" and "depth first". In breadth first a whole ensemble of configurations is obtained at each intermediate stage before proceeding to the next one. In depth first only one full configuration is grown at a time. Both implementations have their own advantages and can be better suited for a particular resampling scheme, etc.

The implementation of the growth algorithm is a hybrid between breadth and depth. It is a hybrid because we grow a whole ensemble at once (typically 10^5 configurations). However, to achieve a good statistical precision we repeat the whole growth process many times and simply combine configurations, energies and weights from different simulations into one large ensemble. Specifically, we used 10 repeats for Ace-$(Ala)_4$-Nme, 100 for Ace-$(Ala)_6$-Nme, and 1000 for Ace-$(Ala)_8$-Nme and Met-enkephalin. The approximate ensembles

for Ace-$(Ala)_{12}$-Nme and Ace-$(Ala)_{16}$-Nme were generated using one repeat.

D. RESULTS

We applied the polymer-growth algorithm to equilibrium sampling of several systems from octane C_8H_{18} and cetane $C_{16}H_{34}$ to peptides including Ace-$(Ala)_4$-Nme, and Ace-$(Ala)_6$-Nme, Ace-$(Ala)_8$-Nme and Met-enkephalin. We study a standard all–atom octane and cetane model using the OPLSAA forcefield [15]. For each system, a 10 nsec dynamical trajectory is generated at 298K using Langevin dynamics (as implemented in Tinker v.4.2.2) in vacuum with friction constant 91/ps. We join two butane ensembles into an octane ensemble, and further combined two octane ensembles into a cetane ensemble.

Then we used a simple solvent model with uniform dielectric constant of 60 at 298 K. The dielectric constant was chosen based on several trial simulations to give reasonable agreement for Ramachandran propensities [83] with implicit solvation simulations. As discussed in Sec. II.C.6, Ace-$(Ala)_4$-Nme was run for 10 repeated simulations resulting in 10^5 saved structures, Ace-$(Ala)_6$-Nme was run for 100 repeats leading to 10^6 configurations. Ace-$(Ala)_8$-Nme and Met-enkephalin were run for 1000 repeats also resulting in 10^6 saved configurations.

To compare the growth results we ran standard Langevin dynamics simulations at atomistic level for the same two alkane systems and four peptides described by the same forcefield and solvent model. Specifically, all systems were sampled for 200 ns at the temperature of 298 K and the friction constant of 91/ps for alkane and 5/ps for peptides. The Langevin dynamics was used as implemented in Tinker software package. All growth and Langevin dynamics simulations were performed on a single Xeon 3.6 GHz CPU and 2 GB of system memory.

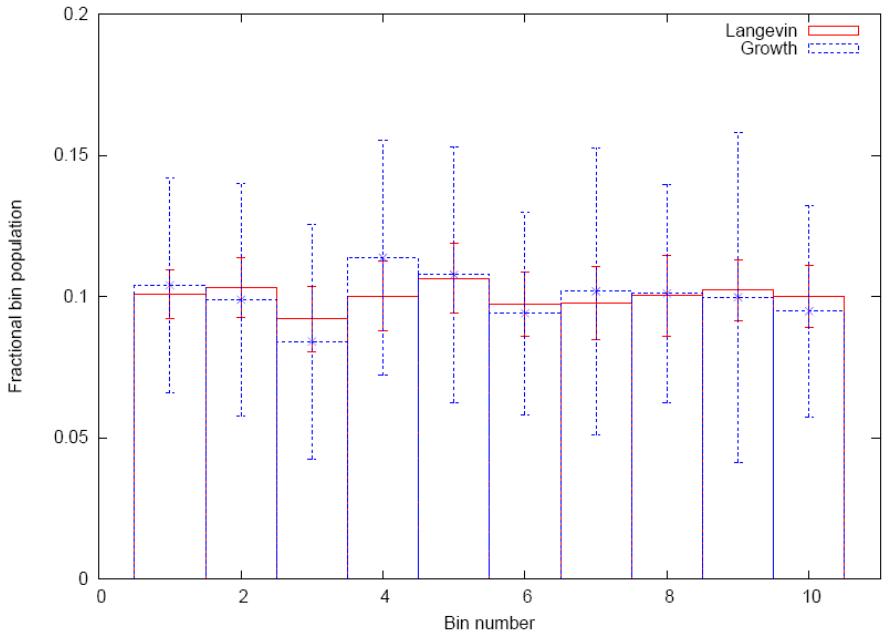

Figure 4: Fractional population of 10 Voronoi bins constructed from growth and Langevin simulations for octane. Error bars represent two standard deviation for each bin, based on 20 independent simulations for both Langevin and growth.

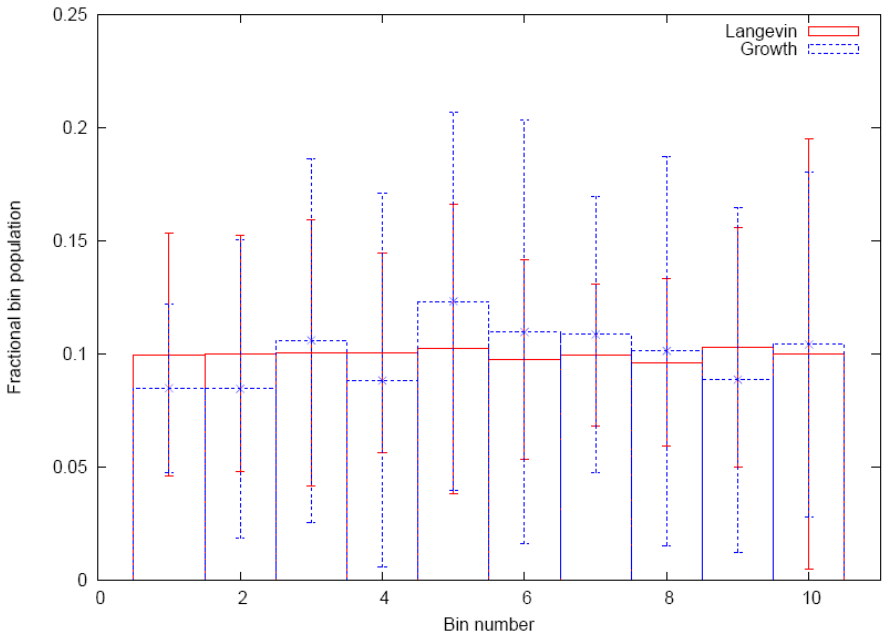

Figure 5: Fractional population of 10 Voronoi bins constructed from growth and Langevin simulations for cetane. Error bars represent two standard deviation for each bin, based on 20 independent simulations for both Langevin and growth.

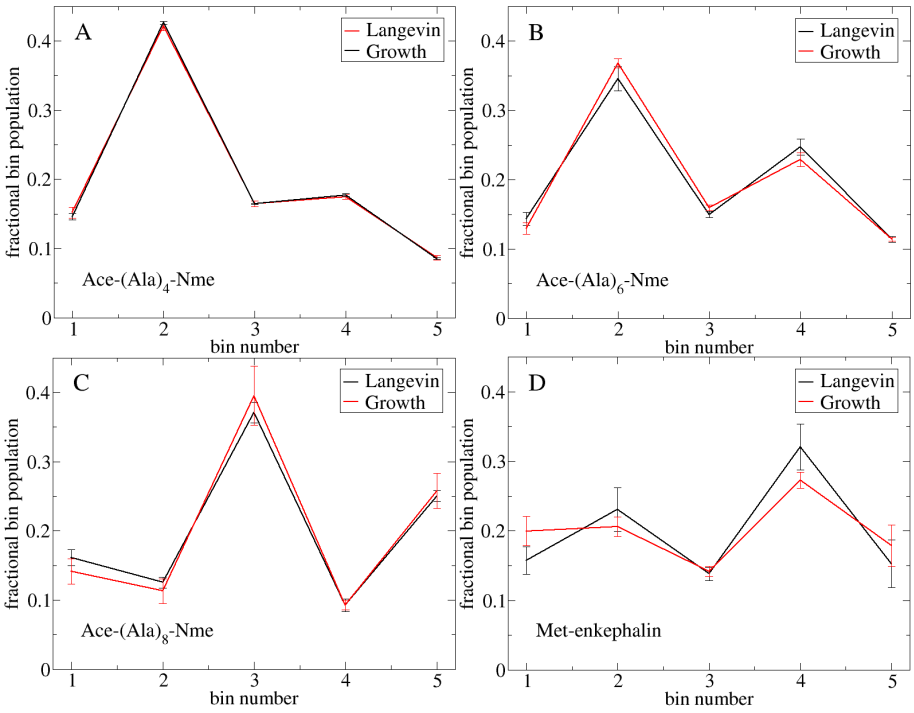

Figure 6: Fractional population of Voronoi bins constructed from growth and Langevin simulations for four peptides: (A) Ace-$(Ala)_4$-Nme, (B) Ace-$(Ala)_6$-Nme, (C) Ace-$(Ala)_8$-Nme, and (D) Met-enkephalin. The bins were constructed based on a Voronoi classification of configuration space. Error bars represent one standard deviation for each bin, based on 12 independent simulations for both Langevin and growth.

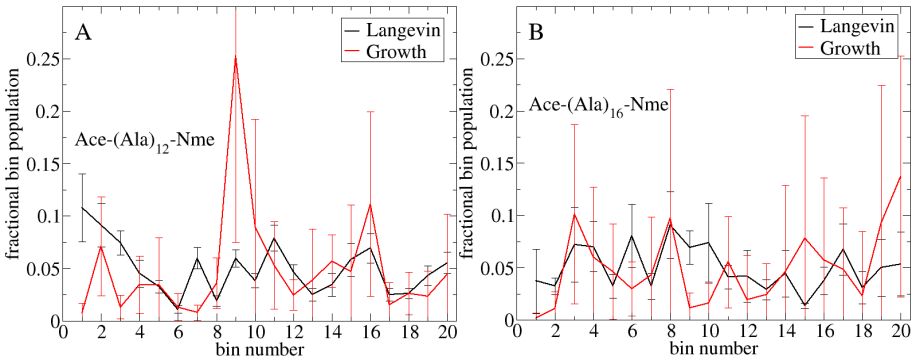

Figure 7: Fractional populations of Voronoi bins constructed from approximate growth procedure and Langevin simulations for two peptides: (A) Ace-$(Ala)_{12}$-Nme, and (B) Ace-$(Ala)_{16}$-Nme. The bins were constructed based on a Voronoi classification of configuration space.Error bars represent one standard deviation for each bin, based on 12 independent simulations for growth and 10 for Langevin.

We first checked that the algorithm can correctly sample the equilibrium distributions by comparing it with Langevin dynamics in which all the parameters are measured from experiments. The equilibrium distributions were compared using structural histograms constructed using Voronoi procedure as described in previous work. The results are shown in Fig. 4 and 5 for octane and cetane system, and Fig. 6 for peptides systems.

These figures indicate mostly good agreement between the two methods – although there appears to be slight deviation in the Met-enkephalin results: see Sec. II.E.

To assess the efficiency of growth simulations we calculated the effective sample size (ESS) of Langevin simulations using two different statistical tools described in Sec. II.B.6. The first method is based on calculating the variance in the approximate physical states, which will be discussed in Chapter IV. The second method employs previously developed de-correlation time analysis by Lyman and Zuckerman [26] and was used to check the results of the first method. The results are reported in Table 1 and indicate a reasonable agreement between two statistical methods. We note that the de-correlation time analysis can be used

system	Number of atoms	$t_{Langevin}$	$ESS_{Langevin}$ from physical states analysis	$ESS_{Langevin}$ from decorrelation analysis
Ace-$(Ala)_4$-Nme	52	28h	2180	2500
Ace-$(Ala)_6$-Nme	72	48.3h	615	800
Ace-$(Ala)_8$-Nme	92	76h	385	330
Met-enkephalin	84	80h	55	130

Table 1: The results of statistical analysis of Langevin dynamics simulations are reported for four peptides. The effective sample size ($ESS_{Langevin}$) was calculated using two different statistical tools as described in Sec. II.B.6

only for dynamic trajectories and, therefore, was not used for growth simulations.

The ESS of growth simulations was calculated using the first statistical tool *i.e.*, by computing the variance in the populations of approximate physical states. The results of this analysis are reported in Table 2 and indicate that a large efficiency gain of over one order of magnitude was achieved for most peptides.

We emphasize that the efficiency of polymer growth algorithms applied to atomistic forcefields at 298 K depends significantly on implementation. In fact the original, naive implementation was not efficient at all – it was several times slower than Langevin simulations. However, in a series of implementation improvements described in Sec. II.C, we achieved good efficiency.

To aid future research in the field, we report how different improvements contributed to the efficiency of growing the peptide Ace-$(Ala)_4$-Nme. The largest improvement, of about two orders of magnitude, can be attributed to using Cartesian coordinates and recycling energy terms. Software optimizations improved the efficiency by about three times. Implementation of the optimal resampling algorithm increased the efficiency by almost another order of magnitude.

system	Number of Atoms	Number of Fragments	t_{Growth}	ESS_{Growth}	γ
Ace-$(Ala)_4$-Nme	52	6	1min	2800	2150
Ace-$(Ala)_6$-Nme	72	8	10.6min	170	75
Ace-$(Ala)_8$-Nme	92	10	1.75h	45	5
Met-enkephalin	84	7	1.5h	100	100

Table 2: The results of the statistical analysis of growth simulations are reported for four peptides. The effective sample size (ESS_{Growth}) was obtained based on calculating the variance in the approximate physical states as described in Sec. II.B.6. The efficiency gain γ relative to Langevin dynamics was calculated using Eq. II.13 . Note that γ was obtained using $ESS_{Langevin}$ calculated from the variance in the physical states.

Besides equilibrium sampling of small peptides, the growth procedure can be also used for rapid generation of approximate ensembles (*i.e.,* steric-clash free) of larger peptides. We generated approximate ensembles for Ace-$(Ala)_{12}$-Nme and Ace-$(Ala)_{16}$-Nme peptides. Each required less than a minute of a single-processor wallclock time. To check whether these approximate ensembles provide reasonable distributions in configuration space, we compared them with equilibrium ensembles from Langevin simulations. The distributions were compared based on structural histograms constructed using a Voronoi procedure. A larger number of bins (20) were used to investigate whether reasonable coverage of configuration space was obtained; the issue of coverage could be important in fragment-assembly applications. The results for both peptides are shown in Fig. 7 and indicate that, indeed, the approximate ensembles yield reasonable coverage of configuration space.

E. DISCUSSION

1. Limitation

The key limitation of the present implementation of the growth algorithm is that it can be applied for precise equilibrium sampling only of relatively small peptides. The limit is about eight amino acid residues and less than 100 atoms, although we showed that significantly larger peptides can be treated approximately. The size limitation for precise sampling appears to result from the small overlap between non-interacting and fully interacting fragment distributions. In the present implementation, the overlap significantly decreases with system size because configurations which predominate in the partially grown ensemble may be less important in the full molecule. For example, if one is growing a largely helical peptide, partially grown configurations will not "know" about hydrogen bonds which will be formed later in the growth process. In Sec. II.E.3 we describe possible solutions to the problem of small overlap.

The present procedure is also limited to implicit solvent models. It is not clear whether explicit solvent could be treated practically, but as discussed below, it should be possible in principle to grow explicit solvent.

2. Possible applications

There are numerous applications for any fast peptide sampling scheme, and some that are specific to the growth scheme. First, it is important to recognize that relatively small peptides have important functions as hormones, neurotransmitters, antigens, and drugs [84]. Examples include enkephalins and vasopressin, just to name a few. Pharmacologically, peptides are actively being investigated as potential drugs [85]. Fast growth procedures can permit the investigation of large numbers of sequences.

Unlike dynamic methods, the polymer growth approach can be used to calculate the absolute free energy without any additional cost [69]. This is possible because all the required generating probabilities and weights are known [86]. In the previous study [69], we calculated absolute free energies for several peptides based on pre-calculated molecular fragments;

however, that study did not employ the critical efficiency improvements described here. In principle, different peptides (or other molecules) can be grown in the presence and absence of a protein to yield relative binding affinities via a standard thermodynamic cycle.

The approximate growth procedure could be of particular use in conjunction with fragment assembly protocols for protein structure prediction [74–76]. Presently, these protocols rely on expensive dynamic sampling of fragment configurations for subsequent assembly into native-like structures. The growth procedure can rapidly generate approximate ensembles of peptides suitable for such assembly or perhaps with other fragment-based programs like Rosetta [66, 87].

Interestingly, the growth procedure can be adapted to multi-resolution sampling because of flexibility in how a molecule is divided into fragments. For example, fragments can correspond to the backbone and side chains of different types. In such a version of the growth algorithm – that we will call "decorating" – given a backbone ensemble, side chains can be added one at a time. Decorating is a true multi-resolution technique because the backbone can be sampled using other canonical methods, for example, the previously developed library-based Monte Carlo. Initial data obtained from decorating (not shown) suggest it can be a useful approach.

3. Possible improvements

There are several possible solutions to the problem of small overlap. One possibility is to bias the growth based on some prior knowledge of the full molecule's configuration-space distribution. This knowledge may be obtained from previous dynamics or growth simulations even if these simulations are not fully sampled, provided there is some information on correlations among all fragments. For example, the biasing can be implemented as a "self-guided" algorithm: a regular growth simulation can be performed first and then subsequent growth simulations can be biased toward important parts of configuration space based on the information obtained in the first simulation. Libraries could also be biased based on simulations and/or databases like the protein data bank.

Efficiency for larger systems might be improved by expanding the ensemble at every

intermediate stage j. For instance, ensemble expansion could be effected using local "relaxation" of the growing configurations with a canonical sampling method, such as library-based Monte Carlo. This idea is based on "annealed importance sampling" [88, 89]. An enlarged canonical ensemble at stage j should have more configurations pertinent to stage $j+1$. In general, growth and dynamic approaches have features that can help each other to better sample configuration space. Growth can instantaneously cross potential energy barriers but is not good at exploring local configuration space. On the other hand, relaxation of partially grown configurations may help to remove strains and better explore local configuration space. Canonical relaxation preserves the correct distribution [88–90].

It is natural to consider fragments larger than those used here, which were single amino acids. For instance, fragments can correspond to amino-acid dimers or even larger peptides. There are two practical limitations on fragment size, however, both of which will become less severe with increasing memory. One restriction stems from sequence variations within a fragment. For example, for dimer fragments it will be necessary to generate and store at least $\frac{20*19}{2} = 190$ different libraries. Another practical limitation is the number of configurations required to adequately describe each library. The present procedure employs 10^5 configurations per library but larger fragments may require significantly larger libraries. On the other hand, biasing libraries toward the most pertinent parts of configuration space will decrease the storage requirements. Again, as computer memory increases and becomes more affordable, these limitations may become less important.

Despite these limitations we tested the potential of using larger fragments in the growth procedure. We employed $(\text{Ala})_2$ and $(\text{Ala})_3$ fragment libraries each containing 10^5 configurations to sample Ace-$(\text{Ala})_8$-Nme but found that the efficiency was inferior compared to using a single Ala fragment. One reason why larger fragments did not help may be that they require much larger libraries (compared to single-residue fragments) to represent their large configuration space.

In this initial study we employed a simple solvent model with uniform dielectric although more accurate models such as GBSA can be implemented. When using a new solvent model, fragment libraries will have to be regenerated although it requires only one time cost. Additional energy terms for the solvent model will have to be implemented in the algorithm.

In principle, polymer growth algorithms are not limited to implicit solvent models. Similar to growing peptides, water molecules can be added one at a time to solvate the system. In fact, Zuckerman group has already "grown" a simple Lennard-Jones fluid [91].

The polymer growth algorithms are well suited for modern graphics processing units (GPUs) because multiple configurations can be grown at once in contrast to dynamic simulations where only one configuration can be processed at a time. The advantage of GPUs is that they have hundreds of arithmetic units where multiple interactions and/or configurations can be simultaneously processed.

III. APPLICATION OF THE SAMPLING METHOD IN ABSOLUTE FREE ENERGY CALCULATION

A. OVERVIEW

In this chapter, I will introduce an application of growth algorithm that was discussed in the previous chapter in absolute free energy calculation. I employ such fragment libraries and interaction tables for amino acids and capping groups to estimate free energies for small peptides. Equilibrium ensembles for the molecules are generated at no additional computational cost, and are used to check results by comparison to standard dynamics simulation. I will also explain how this work can be extended to estimate relative binding affinities. This chapter has been adapted from a published paper [69].

1. Historical background of free energy calculation

Free energy difference ΔF calculations between two ensembles or physical states are useful for a wide variety of applications, including drug design [92] and protein/ligand binding affinities [93, 94]. Free energy difference methods can be classified as either equilibrium or non-equilibrium. Equilibrium approaches rely on fully sampled equilibrium simulations performed at each stage of the free energy calculation. In other words, if the equilibrium is not obtained at any stage of the free energy calculation, a bias will be introduced. There has been extensive study of non-equilibrium methods since the application of Jarzynski's remarkable equality [95, 96]. Non-equilibrium methods have the potential to provide very rapid estimates of free energy difference, but can also suffer from significant bias.

Free energy differences could be easily estimated if one knows the absolute free energy

of the two states or systems. Ytreberg and Zuckerman developed a method to calculate absolute free energy by estimating free energy difference of two systems [97], one of which is reference system with zero free energy. The work introduced in this chapter is an extension of their work, by applying the growth algorithm discussed in the previous chapter.

2. The use of reference systems for free energy calculation

The use of a reference system for free energy calculations has a long history in physics and chemistry. [98] The basic idea is to employ a reference system ("ref") for which the absolute free energy is available, and which is as similar as possible to the physical system of interest ("phys"). Historically, Stoessel and Nowak applied the reference-system strategy to a molecular system for the first time, using a solid harmonic reference system in Cartesian coordinates. [99] Zuckerman and Ytreberg extended that work in two ways designed to improve overlap between the reference and physical systems: [97] (i) by using internal coordinates; and (ii) by using a more flexible, numerically exact reference system based on histograms from a short dynamics simulation, rather than an artificial analytically tractable reference state. Huang and Makarov also employed the reference-system approach embodied in Eq. III.1, but in a different way [100]

$$F^{\text{phys}} = F^{\text{ref}} + \Delta F_{\text{ref}\rightarrow\text{phys}} \ , \tag{III.1}$$

where F^{x} is the absolute free energy of model "x," and $\Delta F_{\text{ref}\rightarrow\text{phys}}$ is the free energy difference between the systems. In essence, this paper is about practical choices for the both the reference system and strategies for calculating $\Delta F_{\text{ref}\rightarrow\text{phys}}$ when the physical system is a large molecule.

Other efforts to calculate absolute free energies for molecular systems have been ongoing for years in the groups of Meirovitch [101–104] and Gilson [105–108] and more approximately using harmonic and quasi-harmonic methods. [109] The work of Meirovitch builds on long-standing polymer-growth methodologies for estimating partition functions which date to the work of Rosenbluth and Rosenbluth. [110] The original Rosenbluth work was generalized for higher efficiency and more realistic models by many workers. [42, 44–47, 49–54, 111, 112] Ideas from these polymer-growth sampling methods also inform the present work.

The present work significantly extends the previous work by Ytreberg and Zuckerman [97] by estimating absolute free energies for molecules built up gradually from molecular fragments. Larger molecules can be treated, compared to the previous paper [97], because a series of staged intermediate systems are adopted. In essence the free energy difference of Eq. III.1 is sub-divided (staged) into a sum of easy-to-calculate terms. Staging increments are highly tunable, based on the choice of fragment sizes and even by selection of subsets of interactions as detailed below. The use of fragments in other types of molecular mechanics calculations has a long history. [8, 78, 113]

A key contribution of this work is a practical strategy of pre-calculation which minimizes the number of energy terms which need to be computed at each stage. Specifically, for each fragment, a statistical library — i.e., an ensemble of configurations and their energies — is stored; we have also used such libraries in Monte Carlo sampling [33]. Additionally, for each covalently bonded fragment pair, we store the full interaction energy (based on all atoms) for every possible pair of configurations. Such storage is quite practical on typical modern computers with more than 1 GB of RAM. During production simulations it is only necessary to compute interactions between fragments separated by one or more other fragments. Needless to say, the stored libraries and interaction tables can be re-used in future simulations of the same or different molecules. The pre-calculation strategy, which has early conceptual roots [43, 79, 80], appears to represent a significant practical advance over earlier polymer-growth calculations. The use of non-Boltzmann distributed libraries has been popularized in the Rosetta protein folding program. [66]

There is substantial flexibility in the division of a molecule into fragments. We have used single amino acids as fragments in this study, but larger segments and even different interaction subsets as detailed below – may also be practical. The fragment-based approach could also be used to study protein-ligand binding, by growing small molecules into receptor binding pockets and estimating the free energies. This can be seen as a statistical mechanical generalization of fragment-based ideas developed earlier. [8, 78]

The results, which employ single amino-acid fragments, are extremely encouraging. The data indicate that absolute free energies for small peptides can be calculated rapidly and reliably. Specifically, high-precision free energy estimates, with fluctuations of ~ 0.3 kcal/-

mole out of 159 kcal/mole, are obtained for 52-atom tetra-alanine in less than an hour of single-processor computing time, with a simple dielectric "solvent". We check the data by comparing the equilibrium ensembles (obtained simultaneously with the free energy estimates) with independent Langevin simulations. As a further check, in one case, the free energy results are verified by an independent calculation using different fragments.

The remaining sections of the chapter describe the methods, results, and conclusions. The methods section provides full details for performing the calculations, including the generation of fragment libraries and interaction tables. I also correct a technical error in the earlier study. [97] The results describe both the free energy values and the analysis of the equilibrium ensembles. The discussion section describes possible improvements to the method and extension to the estimation of relative binding affinities using absolute free energies.

In this project, Dr. Artem Mamonov, a postdoc in the Zuckerman group, generated the libraries and interaction tables. My contribution has been to re-organize the libraries, code the algorithm, perform all the free energy calculations and some optimization.

B. METHODS

The basic approach is to calculate the free energy of the physical system of interest based on the difference from a known reference system, as in Eq. (III.1), and also to stage the calculation using molecular fragments.The fragments not only permit the gradual staging of the calculation but also a tremendous savings of computer time based on the storage of (i) fragment configurations, (ii) energies internal to each fragment configuration, and (iii) interaction energies between covalently bonded fragments. The low cost and high precision of the resulting estimates suggests we are far from the practical limit of the approach in the present implementation. However, a number of improvements to the implementation appear to be within easy reach, as described in the Discussion (see Sec. III.D). All fragment libraries used in the present calculations are available at the website (www.ccbb.pitt.edu/Zuckerman).

1. Model and systems

All calculations employ a standard atomistic forcefield, OPLS-AA [15] at $T = 298K$. In this thesis, the fragments are individual amino acids and capping groups. For simplicity in this initial investigation, we model the solvent by a simple uniform dielectric constant $\epsilon = 60$. We compute free energy estimates for alanine dipeptide (Ace-Ala-Nme), di-alanine (Ace-$(Ala)_2$-Nme), and tetra-alanine (Ace-$(Ala)_4$-Nme). Following standard conventions, Ace is Acetyl (CH_3-CO), Ala is Alanine (NH-CH(CH_3)-CO), and Nme is N-methylamide (NH-CH_3).

2. A simple example

Consider the calculation of the configurational free energy of alanine dipeptide based on a division into three fragments (Ace, Ala, Nme) which can be denoted (A, B, C) respectively (see Fig. 8). In advance, we calculate statistical libraries of configurations for each fragment, which are constant-temperature OPLS-AA ensembles based only on the atoms within the given fragment. The libraries additionally include the six degrees of freedom necessary for joining the fragments, based on the use of "dummy atoms" as described below. During the library generation process, the absolute free energy for each fragment is also calculated using a reference system as described previously. [97] A typical library will contain 10,000 configurations. We also pre-calculate every possible interaction energy between covalently bound fragments — i.e., a table of 10^8 interaction energies (each pair from two 10,000-configuration libraries) for the A-B and B-C fragment pairs.

The calculation proceeds as schematized in Fig. 8, where the presence of a line connecting two fragments indicates that all interactions between the fragments are included. The reference system (not shown) consists of fully independent coordinates, so that the fragments are not yet constructed. The first intermediate consists of the three non-interacting fragments, which include, however, all interactions *within* each fragment. Thus the fragment free energies, which are calculated and stored in advance, properly include the interactions among all degrees of freedom *internal* to each fragment. Other interactions are added in three stages: A-B interactions first, followed by B-C, and completed by A-C couplings. Note that the calculation is under an assumption that there is no many body effects.

In the first intermediate stage, the absolute free energies for the individual fragments are retrieved from disk. (They are initially calculated following reference [97] as detailed below.) Next, A-B interactions are added by a standard free energy difference calculation. Specifically, an ensemble of non-interacting A-B configurations is generated by random combination of fragments from the A and B libraries, and the resulting energy change is exponentially averaged in the usual way — via Eq. (6) below. The energy changes due to the combination do not need to be calculated in this scheme, however, because they have been tabulated in advance. Additionally, the now interacting A-B fragments are "resampled" [34] to correspond to the full potential for all degrees of freedom in *both* fragments. The details of resampling are given below — see Eq. (III.7) — but the bottom line is that one obtains 10,000-configuration ensemble of the partially grown molecule consisting of the A-B fragments.

The calculations then proceeds as if there are two fragments, A-B and C. The two libraries are joined combinatorially *but only accounting for the B-C interactions at this stage.* The A-C interactions will be handled at a later stage. Once again, the free energy change is calculated and the ensemble is resampled to reflect B-C interactions. The resulting ensemble contains 10,000 configurations of the full molecule reflecting all interactions except those between fragments A and C.

In the final stage sketched in Fig. 8, the A-C interactions are added in a standard free energy difference calculation based on the the ensemble of the previous stage. However, the energies are not saved in advance. Rather, my code is able to compute energy terms specific to the A and C fragments — i.e., electrostatic and van der Waals interactions between atoms of A and those of C. Once the energy changes have been obtained, the full free energy is rapidly estimated. Resampling into the fully interacting ensemble can also be performed rapidly without additional energy calculations.

It is not difficult to imagine generalizing this example to systems with more fragments. It is also worth noting that, strictly speaking, the final stage was not necessary. That is, we could have added the A-C interactions simultaneously with the B-C combination since the full molecular configurations were constructed at that point. These choices illustrate the flexibility intrinsic to staging the calculation with fragments, as I detail further in the

Sec. III.D. Additional staging flexibility results, of course, from the initial choice of the fragments — i.e., smaller fragments lead to staging in finer increments.

3. Basic formalism

The calculation of the absolute free energy $F^{\rm phys}$ for a molecule divided into fragments is based on standard, straightforward equations. The only novel aspect of the formalism is the particular choice of stages based on the addition of fragments and/or inter-fragment interactions. Although the heavy reliance on pre-calculated information has very significant practical implications, it does not affect the formalism.

a. Definition of the free energy The fundamental object of interest is the absolute classical free energy $F^{\rm phys}$ for an implicitly solvated molecule. The molecule is taken to consist of N atoms, and its *internal*-coordinate configurations are denoted by $\mathbf{x}$. The potential energy function will be a standard forcefield (here, OPLS-AA [15]), possibly augmented by an implicit solvation model; the full potential energy including any solvation will be denoted by $U^{\rm phys}(\mathbf{x})$. The free energy, which is a functional of $U^{\rm phys}$, is defined by the dimensionless *configurational* partition function at temperature $T = 1/k_B\beta$ via

$$F^{\rm phys}[U^{\rm phys}] = -k_B T \ln \left\{ \frac{1}{\left(1\ \text{Å}\right)^{3N-3}} \int d\mathbf{x}\, e^{-\beta U^{\rm phys}(\mathbf{x})} \right\} , \qquad \text{(III.2)}$$

where the measure of integration $d\mathbf{x}$ is understood to include any necessary Jacobians. Kinetic energy terms have already been integrated out. Both the dimensionless character of the partition function in Eq. (III.2) and the angstrom-based normalization result from a particular choice for the standard concentration C° (defined in references [97, 106]) — or equivalently, for the volume containing the implicitly solvated molecule. C° is built in the kinetic energy, and is picked in order to integrate out the kinetic term. That is why C° appears not in $F^{\rm phys}$. In particular, we have chosen $C^\circ \equiv 8\pi^2(1\ \text{Å})^{3N-3} Q_p/\sigma$, where $Q_p = \prod_{i=1}^{N} (2\pi\, m_i\, k_B T/h^2)^{3/2}$ results from the momentum integrals, h is Planck's constant, and σ is the molecule's symmetry number (the number of indistinguishable orientations that a molecule can exhibit by being rotated around symmetry axis). Note that the chosen

standard concentration varies based on the molecule (i.e., based on the number and masses of its atoms), and also eliminates the temperature dependence of Q_p. However, in almost every application of interest (see Sec. III.D), the absolute free energy calculated here ultimately will be used to estimate a free energy difference and eliminate any artifacts due to C°.

The single-molecule formulation, as noted, allows for "implicit solvation" using an effective solvent term in $U^{\rm phys}$ that is solely a function of the internal molecular coordinates $\mathbf{x}$. The present calculations employ a simple uniform dielectric constant ($\epsilon = 60$). In the Sec. III.D, I address the minor technical issues involved with using a more realistic implicit solvent model.

One issue of dimensionality is worth emphasizing. Although there are $3N-6$ internal coordinates for a molecule consisting of N atoms, the integral of Eq. (III.2) has dimensionality of length to the power $3N-3$. This is because $N-1$ bond lengths remain in the full set of internal coordinates $\mathbf{x}$, each of which contributes three powers of length. Put another way, of the six excluded rigid-body/center-of-mass coordinates, the three orientation angles are dimensionless; more specifically, the angles integrate to the factor of $8\pi^2$ included in the definition of C°

b. Staging the free energy calculation As illustrated in the example of Sec. III.B.2, we will calculate the free energy in a series of stages. These can be understood most easily by adding and subtracting the free energies corresponding to k intermediate models,

$$\begin{aligned} F^{\rm phys} &= \left(F^{\rm phys} - F_k\right) + \left(F_k - F_{k-1}\right) + \cdots + \left(F_1 - F^{\rm ref}\right) + F^{\rm ref} && \text{(III.3)} \\ &= \Delta F_{k\to{\rm phys}} + \Delta F_k + \cdots + \Delta F_1 + F^{\rm ref}\ , && \text{(III.4)} \end{aligned}$$

where $\Delta F_j = F_j - F_{j-1}$ and $F_j[U_j]$ is defined in analogy to Eq. (III.2) for the intermediate models defined by U_j. The U_j potentials will be specified below.

All free energy difference calculations will be performed here using the Zwanzig's formulation. [114] Explicitly, for two arbitrary potential energy functions U_a and U_b, one has

$$\begin{aligned} \Delta F_{a\to b} = F_b[U_b] - F_a[U_a] &= -k_BT \ln \left\langle \exp\left[-\beta\left(U_b - U_a\right)\right]\right\rangle_a && \text{(III.5)} \\ &\simeq -k_BT \ln \left\{ \frac{1}{N_a} \sum_{i=1}^{N_a} \exp\left[-\beta\left(U_b(\mathbf{x}_i) - U_a(\mathbf{x}_i)\right)\right] \right\} && \text{(III.6)} \end{aligned}$$

where the subscript a denotes an average performed over configurations distributed according to the U_a ensemble and N_a is the number of configurations in that ensemble. Eq. (III.6) is used to estimate the free energy differences required in Eq. (III.4), and it is exact in the limit $N_a \to \infty$.

We emphasize that succeeding intermediates are constructed to have progressively narrower distributions as more interactions are added, as in the alanine dipeptide example. In other words, we ensure good overlap and reliable ΔF estimates by proceeding in the generalized "insertion" direction. [115, 116]

c. Resampling to obtain staged equilibrium ensembles As the calculation proceeds through the various stages, we will require the correspoinding equilibrium ensembles for each stage, primarily for use in Eq. (III.6). These are obtained by "resampling," the process of converting an ensemble for one distribution into another by eliminating, duplicating, and/or adjusting the weights of configurations in the original distribution. [34] In my case, primarily use elimination of configurations from a larger ensemble (e.g., all combinations of fragments A and B) to create a smaller one (e.g., the interacting A-B ensemble); I do not adjust weights. More specificially, to resample an ensemble of configurations $\mathbf{x}_a$ generated according to U_a into a U_b ensemble, the original configurations are randomly selected with probability proportional to the ratio of Boltzmann factors,

$$e^{-\beta[U_b(\mathbf{x}_a)-U_a(\mathbf{x}_a)]} \,. \tag{III.7}$$

Operationally, we select configurations by forming a cumulative distribution function (cdf) based on the normalized set of ratios (III.7), and then choosing from this cdf as many times as desired.

4. Choice of intermediate models

As already noted, the set of intermediate models $\{U_j\}$ can be chosen in a variety of ways. In the present study, we employ k intermediates for a molecule divided into k fragments. This was exemplified for alanine dipeptide, which is divided into three fragments.

The present study uses a uniform staging strategy for all molecules examined, as exemplified in Figs. 8 and 9. The reference system consists of *all* coordinates fully independent — both within and between fragments — as in the previous work. [97] The first intermediate stage adds interactions *within fragments*, so that one has true molecular fragments but no interactions between fragments. (These are pre-generated and stored in advance) We then add interactions between neighboring, covalently bound fragments — i.e., among all the atoms in the neighboring fragment pair — one fragment pair at a time. The final stage of this simple scheme involves the addition of all remaining interactions, which occur solely between non-adjacent fragments. The result is a molecule with atoms interacting fully according to a standard forcefield and possibly continuum solvent model.

To explicitly illustrate the staging scheme employed here, consider the case of a molecule divided into the three fragments A, B, and C, as in Fig. 8. We denote by $u_i^{\rm ref}$ the reference potential for internal coordinate x_i, where the full set is $\mathbf{x} = (x_1, x_2, \ldots)$. For the fragments, we let U_y be the potential energy for all interactions internal to fragment y, and U_{yz} is the potential energy for all interactions between the y and z fragments. A three-fragment molecule, consisting of N atoms, would be staged as follows:

$$\begin{aligned} U^{\rm ref} &= \sum_{i=1}^{3N-6} u_i^{\rm ref}(x_i) \\ U_1 &= U_A + U_B + U_C \\ U_2 &= U_1 + U_{AB} \\ U_3 &= U_2 + U_{BC} \\ U^{\rm phys} &= U_3 + U_{AC} \ . \end{aligned} \tag{III.8}$$

The choice of the reference potentials $\{u_i^{\rm ref}\}$ is guided by the forcefield, as detailed below in Sec. III.B.7.

A four-fragment molecule, such as di-alanine (Ace-$(Ala)_2$-Nme) schematized in Fig. 9, would be staged according to:

$$\begin{aligned} U^{\text{ref}} &= \sum_{i=1}^{3N-6} u^{\text{ref}}(x_i) \\ U_1 &= U_A + U_B + U_C + U_D \\ U_2 &= U_1 + U_{AB} \\ U_3 &= U_2 + U_{BC} \\ U_4 &= U_3 + U_{CD} \\ U^{\text{phys}} &= U_4 + U_{AC} + U_{BD} + U_{AD} \end{aligned} \tag{III.9}$$

As described in Sec. III.D, it is also possible to stage the final ("non-bonded") pairwise interactions separately.

We anticipate that significant optimization can be obtained by adjusting fragmentation and staging schemes. While in Sec. III.D, below, describes more gradual staging strategies, the present initial report is limited to the single staging strategy given above.

5. The non-interacting reference system

The computation of the (absolute) reference free energy F^{ref} is perhaps the most technically involved step of the calculation. The remaining free energy differences in the decomposition of F^{phys} in Eq. (III.4) are estimated using a simple, standard method. For the reference free energy, however, great care must be taken with the normalization and Jacobian factors of the chosen probability distributions. Indeed, Ref. [97] includes an error in this regard, as explained at the end of this subsection.

As described in the discussion of staging (Sec. III.B.4), the reference system for all molecules studied here consists of the set of non-interacting internal coordinates. The reference potential energy function will be constructed, following previously published work [97], so that the reference partition function is normalized to one. That is, we will *construct* the reference model U^{ref} so that

$$Z^{\text{ref}}[U^{\text{ref}}] = e^{-\beta F^{\text{ref}}} = \frac{1}{\left(1 \text{ Å}\right)^{3N-3}} \int d\mathbf{x}\, e^{-\beta U^{\text{ref}}(\mathbf{x})} \equiv 1 \,, \tag{III.10}$$

where the same standard concentration as in Eq. (III.2) has been used implicitly. (From this point forward, we will omit writing the length units, but they should be implicitly associated with every bond-length integration.) The motivation for the unit normalization of $Z^{\rm ref}$ is that application of a logarithm leads to the simplifying value,

$$F^{\rm ref} \equiv 0 \ , \tag{III.11}$$

for *every* system.

While there are many ways to construct $U^{\rm ref}$ to satisfy the required normalization of Eq. (III.10), we use the strategy of employing independent internal coordinates as in the earlier work. [97] As usual, the full set of $3N-6$ internal coordinates, $\mathbf{x} = (x_1, x_2, \ldots, x_{3N-6})$ consists of $N-1$ bond lengths, $N-2$ bond angles, and $N-3$ dihedrals. So long as the distribution of each individual coordinate is normalized when integrated with the appropriate Jacobian factor J, the full distribution will be normalized.

Because total reference energy is given by a simple sum of independent terms,

$$U^{\rm ref}(\mathbf{x}) = \sum_{i=1}^{3N-6} u_i^{\rm ref}(x_i) \tag{III.12}$$

the desired normalization (III.10) is ensured by enforcing

$$\int dx_i \, J(x_i) e^{-\beta u_i^{\rm ref}(x_i)} = 1 \ , \tag{III.13}$$

where the inclusion of inverse length units is understood for bond-angle integrals. In words, then, each individual potential $u_i^{\rm ref}$ must include suitable normalization — which is accomplished by offsetting the potential by the log of the integrated (un-normalized) Boltzmann factor. See reference [97] for further information.

(As detailed in Sec. III.B.7, peptide ϕ and ψ angles were, in fact, sampled together from a single distribution based on a pairwise energy function $u_{\phi\psi}^{\rm ref}$. These angles are independent from all other coordinates, however. We emphasize that this exception does not alter the basic formalism, which has been simplified very slightly for clarity.)

It is very useful to observe that normalization of the coordinate distributions via Eq. (III.13) can be achieved either using standard analytic forms — e.g., Gaussians — or via

numerical histogramming procedures. [97] Thus, there is great flexibility in the choice reference distributions embodied in the reference potentials. In addition to forcefield terms, prior knowledge, such as from a simulation, can be used in constructing the set $\{u_i^{\rm ref}\}$. The reference potentials chosen for the present study are described below in Sec. III.B.7 on library construction.

One word of warning is appropriate here. Although it is possible to describe the internal configuration of a molecule using additional bond angles to substitute for dihedrals in some cases, the Jacobian for such a description appears *not* to be well-defined. Therefore, it is necessary to use the standard description with $N-2$ bond angles and $N-3$ dihedrals. Unfortunately, this point was handled incorrectly in the original publication [97] and therefore an erratum will be prepared correcting the resulting numerical errors. The correct procedure is used here.

6. First intermediate: non-interacting fragments

The first intermediate stage adds only localized interactions to the non-interacting reference model, as illustrated in Figs. 8 and 9. Specifically, once a molecule is divided into fragments (A, B, C, ...), the first intermediate includes only interactions occuring *within fragments*. The fragments exactly divide all coordinates so that we can write $\mathbf{x} = (\mathbf{x}_A, \mathbf{x}_B, \ldots)$ and the potential energy function for this stage is given by

$$U_1(\mathbf{x}) = U_A(\mathbf{x}_A) + U_B(\mathbf{x}_B) + \cdots \ , \tag{III.14}$$

where U_y includes all interactions from the *full forcefield* (OPLS-AA, in this case) among the fragment coordinates $\mathbf{x}_y$ for $y = A, B, \ldots$. Importantly, the fragment potential U_y includes all non-bonded interactions — electrostatic, van der Waals — among the atoms of the fragment. (Sec. III.B.7 on the libraries describes the treatment of connecting "dummy" atoms.)

The free energy for this stage — i.e., $F_1[U_1]$ for use in the key equation (III.4), recalling $F^{\rm ref} \equiv 0$, can be calculated by using the standard perturbation relation (III.6). For such a computation, one would use $U_a = U^{\rm ref}$ and $U_b = U_1$ along with an ensemble distributed according to the Boltzmann factor of $U^{\rm ref}$.

In practice, *once the libraries are generated, no calculation of energies needs to be done.* As detailed below the libraries are generated (just once, for repeated use in many systems) based on the U^{ref} distribution. Thus, during the library generation process, it is a trivial matter to calculate the absolute free energy for each fragment using Eq. (III.6). Thus, individual fragment free energies F_y are calculated in advance that exactly sum to the desired first-stage free energy:

$$F_1 = F_A + F_B + \cdots \ . \tag{III.15}$$

Further, the independent-coordinate distributions are subsequently resampled based on Eq. (III.7) to generate the library distributions — i.e., ensembles for the U_y Boltzmann factors — for use in subsequent stages.

7. Construction of fragment libraries

As just described, fragment libraries are critical to the calculation of the free energy of the first intermediate stage, F_1. The libraries also greatly facilitate computations for the rest of the intermediates.

In general terms, fragment configurations are generated by sampling internal coordinates according to the independent probability distributions which constitute the reference system. The generated configurations are then used to calculate fragment free energies, F_y for $y = A, B, \ldots$. The configurations are also reweighted into an ensemble distributed according to the full forcefield for $\mathbf{x}_y$, the degrees of freedom internal to fragment y. Typically, such a procedure can be applied only to systems with a sufficiently small number of degrees of freedom. For large systems with enough correlated degrees of freedom, there tends to be insufficient overlap with the reference system of independent coordinates. That is, only a tiny fraction of the reference-distributed configurations will be important in the interacting ensemble. Therefore, the choice of the generating probability is essential for the efficient generation of libraries.

We found that for fragments the size of alanine residues, rather simple generating probability were sufficient for generating tens of thousands of (statistically independent) configurations in weeks of single-CPU time. This is a negligible cost because once a library is

generated, it can be used in multiple simulations.

Different coordinate types are best sampled with different distributions, as is suggested by the forcefield terms. Regardless of the particular choice, the specification of the distribution immediately implies the functional form for the reference potential u_i^{ref} from Eq. (III.13). We found that simple Gaussian distributions, with parameters extracted from a short Langevin simulation, worked well for bond lengths and bond angles. For "stiff" dihedrals, such as those in relatively planar groups (e.g., peptide bond), a Gaussian is also appropriate. For "soft," rotatable dihedrals — such as ϕ, ψ and χ angles in amino acids — we simply extracted histograms from a Langevin simulation of alanine dipeptide, as described in reference [97]. A two-dimensional (correlated) probability function was used for the (ϕ, ψ) dihedral pair, but a one-dimensional distribution was used for all other dihedrals. No matter how the coordinates are sampled, $F^{\text{ref}} = 0$ always holds.

Based on the distributions just described, internal coordinates were sampled independently (except for pairwise sampling of ϕ and ψ dihedrals) using an in-house program written by Dr. Artem Mamonov in C. Generated configurations were saved to disk and converted to Cartesian coordinates. The corresponding forcefield energies for each configuration were calculated using the "analyze" module of the Tinker software package. [117] Based on these values and the known reference energies, the individual fragment free energies were calculated using Eq. (III.6). A simple resampling procedure [34] was used to generate a fragment ensemble distributed according to the forcefield; see Eq. (III.7). Only a small fraction ($\gtrsim 10^{-4}$) of reference-ensemble configurations remain after resampling, requiring extensive sampling of the reference ensemble and weeks of CPU cost, as mentioned earlier.

For this study, we generated libraries consisting of 10,000 configurations. All fragment libraries were sampled according to OPLS-AA forcefield at $T = 298K$, with a simple dielectric constant ($\epsilon = 60$) modeling solvent. The choice of dielectric constant was motivated by the reasonable behavior observed in separate Langevin simulations of poly-alanine systems (data not shown).

As noted earlier, for all possible 10^8 *covalent* (neighboring) pairings of fragments, we also tabulated the interaction energies from the forcefield, accounting for all atoms in the fragment pair. Suitable corrections for dummy atoms (see below) were made. In other words,

for a simple two-fragment system, all interactions are stored.

a. Use of dummy atoms Because fragments are sampled independently from each other, the six degrees of freedom that specify the relative orientation of neighboring fragments are included with the fragments. For this purpose "dummy" atoms are used to provide the extra coordinates. We stress that the use of dummy atoms was implemented carefully to avoid adding additional degrees of freedom (e.g., certain bond lengths and angles). We chose to have the dummy atoms interact with the true fragment atoms for better overlap with subsequent ensembles. Thus, when the fragments are joined, the interaction energies of dummy atoms should be subtracted from the full fragment energy because dummy atoms are replaced with neighboring fragment atoms. (Of course, it is simpler to have non-interacting dummy atoms.) For example, in combination of two butane fragments into an octane, two extra hydrogen atoms are attached on one end of both butane fragments (two hydrogen atoms on last carbon), to make them like real butane (three hydrogen atoms on last carbon). The two hydrogen atoms will be discarded upon combination.

The dummy atoms used at the N-terminus of a fragment are carbonyl C, carbonyl O and terminal alpha-C with valence set to one. The dummy atoms used at the C-terminus are amide N, amide H, and terminal alpha-C with valance also set to one. The dummy atoms were assigned the same forcefield parameters as used in the corresponding fragment atoms.

8. The second and subsequent intermediates: adding neighboring fragment interactions

Returning again to the scheme embodied in Eq. (III.4), as well as in Figs. 8 and 9, the next intermediates add interactions between neighboring fragments. These can be considered the "bonded" interactions in the space of fragments, but non-bonded interactions among all atoms in the neighboring pair are included. Explicitly, the models for the remaining intermediates are described by

$$U_2(\mathbf{x}) = U_A(\mathbf{x}_A) + U_B(\mathbf{x}_B) + \cdots + U_{AB}(\mathbf{x}_A, \mathbf{x}_B) \tag{III.16}$$

$$U_3(\mathbf{x}) = U_2(\mathbf{x}) + U_{BC}(\mathbf{x}_B, \mathbf{x}_C) + \cdots , \tag{III.17}$$

where U_{yz} is the full interaction energy — based on the forcefield and solvent model — between fragments y and z.

Formally, it is clear what needs to be done. The ensemble of the previous stage $j-1$ should be used to calculate ΔF_j using the perturbation relation (III.6) with U_{j-1} and U_j.

Again, however, possession of the libraries and interaction tables leads to dramatic practical implications. For instance, by construction, the energy $U_2 - U_1$ is simply the pre-stored energy U_{AB}; similarly $U_3 - U_2 = U_{BC}$. These tabulated energies are used directly in Eq. (III.6) without the need for additional energy calls. The required ensembles for each stage are generated by the rapid resampling procedure of Eq. (III.7). In this way, one readily generates the free energy differences $\Delta F_2, \Delta F_3, ...$ required for the evaluation of $F^{\rm phys}$ via Eq. (III.4).

Caution is required when the molecule of interest contains repeated fragment pairs. While the same libraries can be used for the repeats, say at intermediate stages j and m, *the corresponding values of* ΔF_j *and* ΔF_m *will be different in general.* To see the reason, consider the case of the tetra-alanine peptide studied below. The term ΔF_2 corresponds to including the interaction of the already combined fragments Ace-Ala with the next Ala. Note that the free energy difference ΔF_2 is calculated via Eq. (III.6) using the Ace-Ala ensemble as the "a" system. By contrast, consider the calculation of ΔF_3 for the addition of the next Ala — now to the Ace-$(Ala)_2$ ensemble. Although the free energy change will be based upon the identical (tabulated) interactions, the associated Boltzmann factors in Eq. (III.6) will be *weighted differently — i.e., occur with different frequencies — due to the differing initial "a" ensembles.* In turn, this will lead to different free energy changes, so that $\Delta F_3 \neq \Delta F_2$. This is the mistake I took when I performed the calculation for the first time.

9. The final free energy difference: non-neighboring interactions

As described in the master scheme of Eq. (III.4) and illustrated in Figs. 8 and 9, the final calculation needed to obtain $F^{\rm phys}$ entails the inclusion of all remaining interactions in the forcefield and solvent model. These interactions, excluded until now, occur between atoms in non-neighboring pairs. As described in Sec. II D, for a molecule of k fragments, the full

physical potential energy function (i.e., the forcefield) can be written as the difference from the final (kth) intermediate:

$$U^{\text{phys}}(\mathbf{x}) = U_k(\mathbf{x}) + \sum_{y..z} U_{yz}(\mathbf{x}_y, \mathbf{x}_z) \ , \tag{III.18}$$

where the sum is over non-neighboring pairs of fragments — i.e., AC, AD, BD, ...

In this case, the necessary energy terms for use in the calculation of $\Delta F_{k\to\text{phys}}$ via Eq. (III.6) must be calculated. They cannot readily be stored in advance, due to the combinatorial explosion of possible configurations. For instance, with libraries of 10^4 configurations, there are 10^{12} possible configurations for three fragments, which is beyond the range of current commerical machines.

10. Generating an equilibrium ensemble without additional energy calls

The physical ensemble, distributed according to the Boltzmann factor of the forcefield, can be generated by resampling the U_k ensemble — the last intermediate — using Eq. (III.7). In this case, the "a" ensemble corresponds to U_k and the "b" ensemble to the full forcefield and (implicit) solvent model. Because all energy terms have already been calculated, no additional energy calls need to be made. The necessary resampling computation is extremely fast compared with preceding stages of the protocol.

11. Checking the code and estimating uncertainty

Although the formalism governing the present study is mostly straightforward, my in-house computer program not only needs to reproduce standard forcefield results, but also requires complicated "dissections" of various subsets of forcefield terms. We therefore performed three types of checks on the code. (i) We checked that the forcefield energy for full molecular configurations exactly reproduces the results reported in Tinker (data not shown). This verifies that we have correctly accounted for the dummy-atom energy terms. (ii) Using the previously developed "structural histograms" for analyzing configuration-space distributions, [25, 26] we checked that the equilibrium ensembles produced during the free energy calculations agree with independent Langevin simulations. This data is shown in the Results

section, and generated as explained below. (iii) Finally, we performed a check to ensure that the final free energy values are independent of the choice of fragments. These data, for two- and three-fragment decompositions of alanine dipeptides is also shown in the Results section.

a. Statistical error Statistical uncertainties were calculated by running 20 independent computations for every free energy value reported. Twice the standard deviation among these 20 values is reported, which quantifies the scale of expected statistical error for a *single* simulation. The repeated simulations were run using 20 independent sets of libraries for the various fragments — i.e., the calculation was started all the way at the beginning in each repeat. However, because the overlap between various stages is the limiting factor in the quality of the free energy results, rather than the fairly large libraries, we anticipate similar error estimates would be obtained for one set of libraries.

b. Analyzing equilibrium ensembles/distributions In two previous studies, [25, 26] we have developed methods for comparing equilibrium distributions for molecular systems of arbitrary complexity. The central idea is to employ a "structural histogram" which simply classifies (divides) configuration space into a number of "bins" (regions). Two correct simulations should yield the same results for the fractional populations of the bins, within statistical error for all bins. (Furthermore, the statistical uncertainty in the population estimates can be used to quantify the "effective sample size".) [26] In the present work, we compare equilibrium distributions from fragment combination and from standard Langevin simulations based on structural histograms. The particular histograms employed in the present study have five bins derived from a Voronoi construction; [77] the reference structures for the Voronoi procedure are derived from the equi-probability scheme described in reference. [26] Although the resulting bins are not exactly equally probable, each is guaranteed to represent a contiguous region in configuration space due to the Voronoi construction.

C. RESULTS

The absolute configurational free energy F^{phys} was calculated for the monomer, dimer, and tetramer alanine peptides: alanine dipeptide (Ace-Ala-Nme), di-alanine (Ace-$(Ala)_2$-Nme), and tetra-alanine (Ace-$(Ala)_4$-Nme). For alanine dipeptide, the free energy was estimated based on two different fragment sets as a check on the code. Twenty independent calculations for every F^{phys} estimate were performed to quantify uncertainty, as described above. Additionally, every free energy calculation also yields an equilibrium ensemble, which is compared to independent Langevin simulations.

The results are very positive in every regard, and rather rapid as reported at the end of this section. The amount of memory used, which is a key to the present calculations, is also reported.

The results reflect the uniform protocol adopted here. First, absolute free energies for non-interacting fragments are calculated. Then free energy changes resulting from interactions among covalently bound fragments are added ("bonded" terms, in the space of fragments), one at a time in sequence. Finally, all remaining interactions are added, which account to ("non-bonded") interactions among all non-sequential fragment atoms. The final free energy values reflect the full OPLS-AA forcefield [15] as implemented in Tinker. [117]

1. Alanine dipeptide using two different fragmentations

Because of the complexity of the fragmentation procedure and the lack of reference standards for absolute free energy values, we wanted to ensure the code and procedure were introducing no artifacts. We were particularly concerned about the interacting dummy atoms which introduce "temporary" energy terms, that must be corrected for properly at every combination stage. We find excellent agreement between free energy estimates based on two- and three-fragment decompositions.

a. "Standard" three-fragment decomposition In the standard decomposition for the present study, we separate peptide and amino acid groups. For alanine dipeptide (AD),

then, the three standard fragments are Ace, Ala, and Nme, and the corresponding stages for the free energy calculation are given in Eq. (III.8). Recalling the convention that $F^{\text{ref}} \equiv 0$, the free energy terms from Eq. (III.4) can be written as

$$\begin{aligned} F^{\text{ref}} &\equiv 0 \\ \Delta F_1 &= F_{\text{Ace}} + F_{\text{Ala}} + F_{\text{Nme}} \\ \Delta F_2 &= \Delta F_{\text{Ace}\to\text{Ala}} \\ \Delta F_3 &= \Delta F_{\text{Ala}\to\text{Nme}} \\ \Delta F_{3\to\text{phys}} &= \Delta F_{\text{nonbonded}} \ . \end{aligned} \tag{III.19}$$

where F_y is the absolute free energy (including dummy atoms) for fragment y and $\Delta F_{x\to y}$ indicates the free energy change of combining fragments x and y (which includes all bonded and non-bonded terms, as well as the correction of dummy terms). Finally, $\Delta F_{\text{nonbonded}}$ denotes the free energy change in going from an ensemble where sequentially separated fragments do not interact to a fully interacting ensemble (in this case, the Ace-Nme interactions are added).

b. Two-fragment decomposition As an alternative decomposition, we used Ace-Ala as one fragment and Nme as the other. Importantly, the Ace-Ala library and absolute free energy were *not* generated from a combination of the two smaller libraries, but instead from a ground-up calculation based on independent coordinates as described in the Sec. III.B.

In this case, then, the free energy terms from Eq. (III.4) become

$$\begin{aligned} F^{\text{ref}} &\equiv 0 \\ \Delta F_1 &= F_{\text{Ace}-\text{Ala}} + F_{\text{Nme}} \\ \Delta F_{1\to phys} &= \Delta F_{\text{Ace}-\text{Ala}\to\text{Nme}} \ , \end{aligned} \tag{III.20}$$

where it is notable that in the two-fragment case, *all* interactions are included in the libraries and interaction tables. In other words, no energy calls at all are needed.

c. Comparison of free energies There is essentially perfect agreement between free energies estimate via the two independent decompositions, which provides a reassuring check on the computer program. The full results are given in Table 3. Notably, the two-fragment decomposition has a higher variance, which probably results from a decreased "precision" in the pre-generated Ace-Ala ensemble. In the composite pre-generated Ace-Ala ensemble, the whole configuration space is represented by 10^4 configurations, whereas when Ace and Ala from separate 10^4-member libraries are combined, there is a much denser coverage of configuration space.

d. Equilibrium ensemble compared to standard simulation The free energy computation produces an equilibrium ensemble through repeated resampling procedures at each stage, as explained in the Methods section (Sec. III.B.10). As a further check on the data, we compare the equilibrium ensembles generated from the fragment combination procedure to those produced by long Langevin simulations performed in Tinker. [117] The results, shown in Fig. 10(a), indicate that the computation is indeed producing correct equilibrium ensembles. The graph shows the populations of different regions of configuration space, which was divided up using a Voronoi procedure explained above (Sec. III.B.11). The alanine dipeptide equilibrium distribution was generated from the three-fragment protocol, and the 1 μsec. Langevin simulation (20*50 nsec) was performed in Tinker using a relaxation rate (see Sec. I.B.3) of 10.0 ps^{-1} at $T = 298K$.

2. Di-alanine

Using the same libraries as for the alanine monomer above, we now calculate the absolute configurational free energy for the di-alanine peptide (Ace-$(Ala)_2$-Nme). The staging used is described in Eq. (III.9), which corresponds to the following free energy terms for use in Eq.

(III.4):

$$
\begin{aligned}
F^{\mathrm{ref}} &\equiv 0 \\
\Delta F_1 &= F_{\mathrm{Ace}} + 2 \cdot F_{\mathrm{Ala}} + F_{\mathrm{Nme}} \\
\Delta F_2 &= \Delta F_{\mathrm{Ace}\to\mathrm{Ala}} \\
\Delta F_3 &= \Delta F_{\mathrm{Ala}\to\mathrm{Ala}} \\
\Delta F_4 &= \Delta F_{\mathrm{Ala}\to\mathrm{Nme}} \\
\Delta F_{4\to\mathrm{phys}} &= \Delta F_{\mathrm{nonbonded}} \ .
\end{aligned}
\tag{III.21}
$$

The free energy values are once again calculated with high precision: fluctuations are a fraction of one kcal/mole. The data for all free energy terms is given Table 4, where we see a significant change in the $\Delta F_{\mathrm{nonbonded}}$ term, reflecting the increased number of attractive interactions in this larger molecule (compared to alanine dipeptide).

Similarly, the agreement among bin populations for di-alanine in Fig. 10(b) is excellent, which provides an independent reason for having confidence in the free energy results. The Langevin simulation for di-alanine was performed with exactly the same parameters as for alanine-dipeptide.

3. Tetra-alanine

The results are of high precision (~ 0.1 kcal/mole standard deviation) for tetra-alanine. The staging follows the standard procedure, with the only subtlety in the present case is that the addition of every Ala residue is different, because the "growing" ensemble is different in every case. Thus we consider the first alanine (Ala1) separate from the second (Ala2), and

so on.

$$
\begin{aligned}
F^{\mathrm{ref}} &\equiv 0 \\
\Delta F_1 &= F_{\mathrm{Ace}} + 4 \cdot F_{\mathrm{Ala}} + F_{\mathrm{Nme}} \\
\Delta F_2 &= \Delta F_{\mathrm{Ace}\rightarrow\mathrm{Ala1}} \\
\Delta F_3 &= \Delta F_{\mathrm{Ala1}\rightarrow\mathrm{Ala2}} \\
\Delta F_4 &= \Delta F_{\mathrm{Ala2}\rightarrow\mathrm{Ala3}} \\
\Delta F_5 &= \Delta F_{\mathrm{Ala3}\rightarrow\mathrm{Ala4}} \\
\Delta F_6 &= \Delta F_{\mathrm{Ala4}\rightarrow\mathrm{Nme}} \\
\Delta F_{6\rightarrow\mathrm{phys}} &= \Delta F_{\mathrm{nonbonded}} \ .
\end{aligned}
\tag{III.22}
$$

The data for each of these terms is given in Table 4. Although the different alanine additions are based on different ensembles, the results show they are statistically indistinguishable in this case. However, the "non-bonded" term $\Delta F_{\mathrm{nonbonded}}$ again is significantly different from the previous molecules, as expected.

In comparing the equilibrium distributions from fragment combination and Langevin simulation, once again there is good statistical agreement. For Langevin simulation of tetra-alanine, all parameters were set as before, except for a friction constant of 5.0 ps^{-1}, which does not alter the equilibrium distribution but facilitate the equilibrium process. The contrast between the large fluctuations in the bin populations p_i and the high precision of F^{phys} in Table 4 reveals an important lesson: sampling is harder than free energy calculation.

4. Timing and memory usage

The calculations were reasonably inexpensive, taking 20 minutes for alanine dipeptide, 30 minutes for di-alanine, and 50 minutes for tetra-alanine using one processor of an Intel Xeon 3.20 GHz machine. Concerning memory, a single library containing 10,000 configurations requires 11 MB for Ala, 12 MB for Ace-Ala complex and 5.7 MB for Ace and Nme. An interaction table containing 10^8 pair-wise interactions uses 1.3 GB.

D. DISCUSSION

1. The overall strategy and results

Overall, the precision of the free energy estimates was very high, which can be attributed to two related factors. First, the ensembles in the reference and intermediate stages were of good statistical quality — i.e., characterized by a large effective sample size (data not shown). [26] Second, there was good overlap between the stages, which indeed contributed to maintaining the effective sample size (See Sec. I.C) throughout the stages. The overlap is present by design, as interactions were always *added* between stages. The addition of interactions or, equivalently, correlations among degrees of freedom is guaranteed to reduce the entropy. [118] This progressive narrowing of configuration space is consistent with Kofke's proposal to calculate free energy differences in the "insertion" direction. [115, 116] For larger systems, however, one expects limitations to maintaining the effective sample size using the present protocol, as explained below.

2. Application of fragment combination for estimating relative protein-ligand affinities

Because the fragment combination procedure can be applied to fragments of small molecules, and not just to peptides as in the present thesis, the approach can be applied to calculate approximate relative affinities. That is, one can grow a ligand into the binding pocket of a protein receptor and calculate its free energy. A number of different approximations can be imagined. Most simply, the receptor can be held rigid and the ligand grown in the fields (van der Waals and electrostatic) of the receptor. In a better approximation, the binding-site side-chains can be grown along with the ligand. One can expect affinities based on such free energy calculations to be superior to their docking counterparts because entropy is included. To produce a relative affinity estimate between two ligands, the respective solvation terms would need to be included as usual. [108]

3. Efficiency of fragment combination for equilibrium sampling

As I have already noted, the fragment combination protocol I have described produces equilibrium ensembles simultaneously with free energy estimates. It is natural to wonder whether such ensembles are produced more efficiently than by standard dynamics simulation especially given that small peptides have been shown to have multi-nanosecond relaxation times. [26] In fact, as I discussed in Chapter II, fragment combination can lead to sampling that is faster by several orders of magnitude. However, somewhat more sophisticated resampling schemes and different fragment sizes are useful in reaching the highest levels of efficiency.

4. Use of implicit solvent models

It is interesting and important to consider the additional costs which would be entailed by using a standard implicit solvent model, such as GBSA. [71] First, both the libraries and the interaction tables would need to be regenerated using the implicit solvent model. Although this could take several weeks of single-CPU time, it needs to be done only a single time for a given model. The second cost is for additional solvent calculations not included in the libraries and tables. I hope to report on the staging and computational expense in a forthcoming publication.

5. Relaxation simulations for large systems

In the protocol employed for this study, the equilibrium ensemble generated at one stage, say j, is used to calculate the incremental free energy difference to the next stage, $j+1$. To continue the process, the ensemble at stage $j+1$ is produced by resampling ensemble j as described in the Sec. III.B. However, it is possible that the resampled ensemble will contain a small number of distinct configurations in an important part of configuration space. Such configurations will have high weight prior to resampling but low weight for the full molecule ensemble, and thus the problem can be diagnosed by noting whether any configurations are resampled multiple times. Clearly, duplicated configurations will not be statistically

independent and lead to increased statistical error in free energy estimates.

One solution to this problem would be to relax duplicated configurations — i.e. to perform short equilibrium simulations to create distinct configurations. The statistical justification of such an approach is somewhat technical [88] and will be described in future work as required.

6. Alternative staging using partial interactions

Additional incremental stages can be added by considering only subsets of interactions. For instance, in the case of di-alanine (Ace-$(Ala)_2$-Nme) which is composed of four fragments (A, B, C, D), there are three sets of non-bonded interactions: AC, AD, and BD. The present implementation adds all three in a single stage, but they could be added one at a time in order to get more accurate free energy estimation. Undoubtedly, in larger systems, such finer staging will be necessary and probably will be required by relaxation of duplicated configurations as just described.

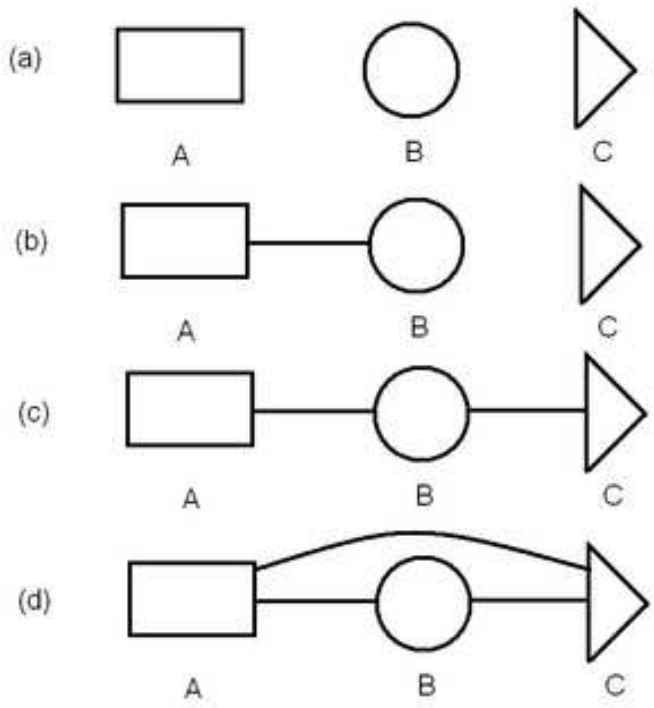

Figure 8: Stages for calculating the absolute free energy of a molecule by combining three fragments, based on Eq. (III.8). Connecting lines schematize full interactions between fragments, including both bonded and non-bonded atomistic terms. (a) The first intermediate stage comprises non-interacting fragments, but includes all interactions *internal* to each fragment. (b) The second stage adds interactions among the atoms of fragments A and B, while (c) the third stage does the same for fragments B and C. (d) In the final stage, representing the desired free energy F^{phys}, all interactions are added, including among non-sequential fragments and possibly including an implicit solvent model.

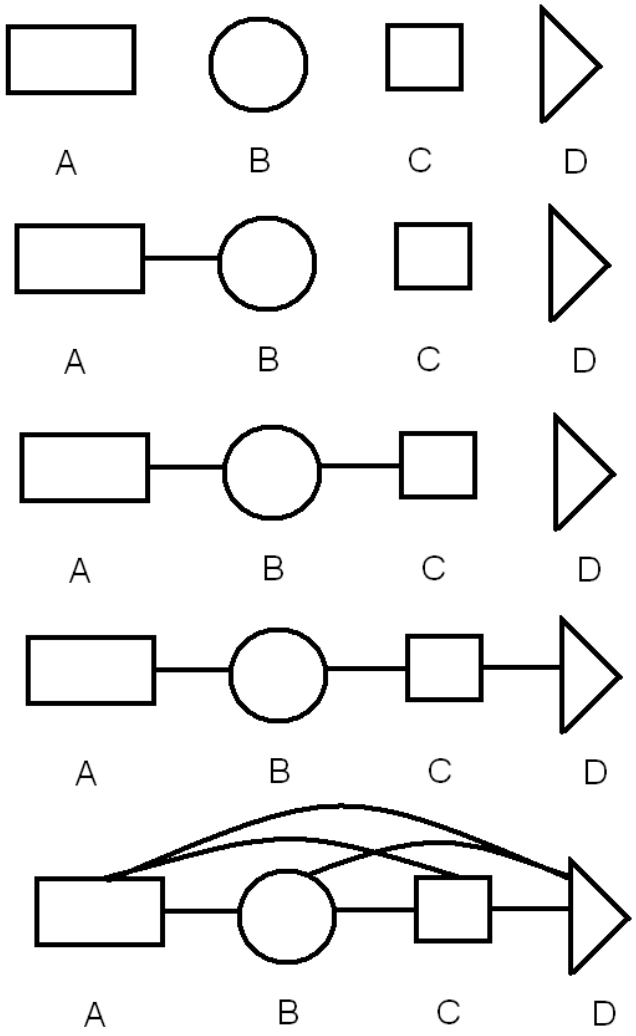

Figure 9: Stages used in the free energy calculation of a four-fragment molecule, corresponding to Eq. (III.9). The initial stages proceed in analogy to Fig. 8, with pair-wise interactions added one at a time for neighboring ("bonded") fragments. In the final stage, *all* remaining interactions are added. Other, more incremental staging schemes are possible, but were not necessary in the present study.

Alanine dipeptide free energy terms from Eqs. (III.19) and (III.20)			
Three Fragments		Two Fragments	
Term	Estimate [kcal/mol]	Term	Estimate [kcal/mol]
F_{Ace}	14.783(0.003)	$F_{\mathrm{Ace-Ala}}$	47.311(0.027)
F_{Ala}	33.326(0.015)	F_{Nme}	16.574(0.003)
F_{Nme}	16.574(0.003)	$\Delta F_{\mathrm{Ace-Ala\to Nme}}$	−0.792(0.002)
$\Delta F_{\mathrm{Ace\to Ala}}$	−0.801(0.002)		
$\Delta F_{\mathrm{Ala\to Nme}}$	−0.499(0.007)		
$\Delta F_{\mathrm{nonbonded}}$	−0.285(0.008)		
F^{phys}	63.098(0.015)	F^{phys}	63.093(0.028)

Table 3: Comparison between the absolute free energy for alanine dipeptide estimate using two different fragmentation schemes. The "standard" three-fragment decomposition (Ace, Ala, Nme) is compared to a two-fragment grouping (Ace-Ala, Nme). The table gives free energy values in kcal/mole, as well as two standard deviations (in parentheses) based on 20 independent calculations.

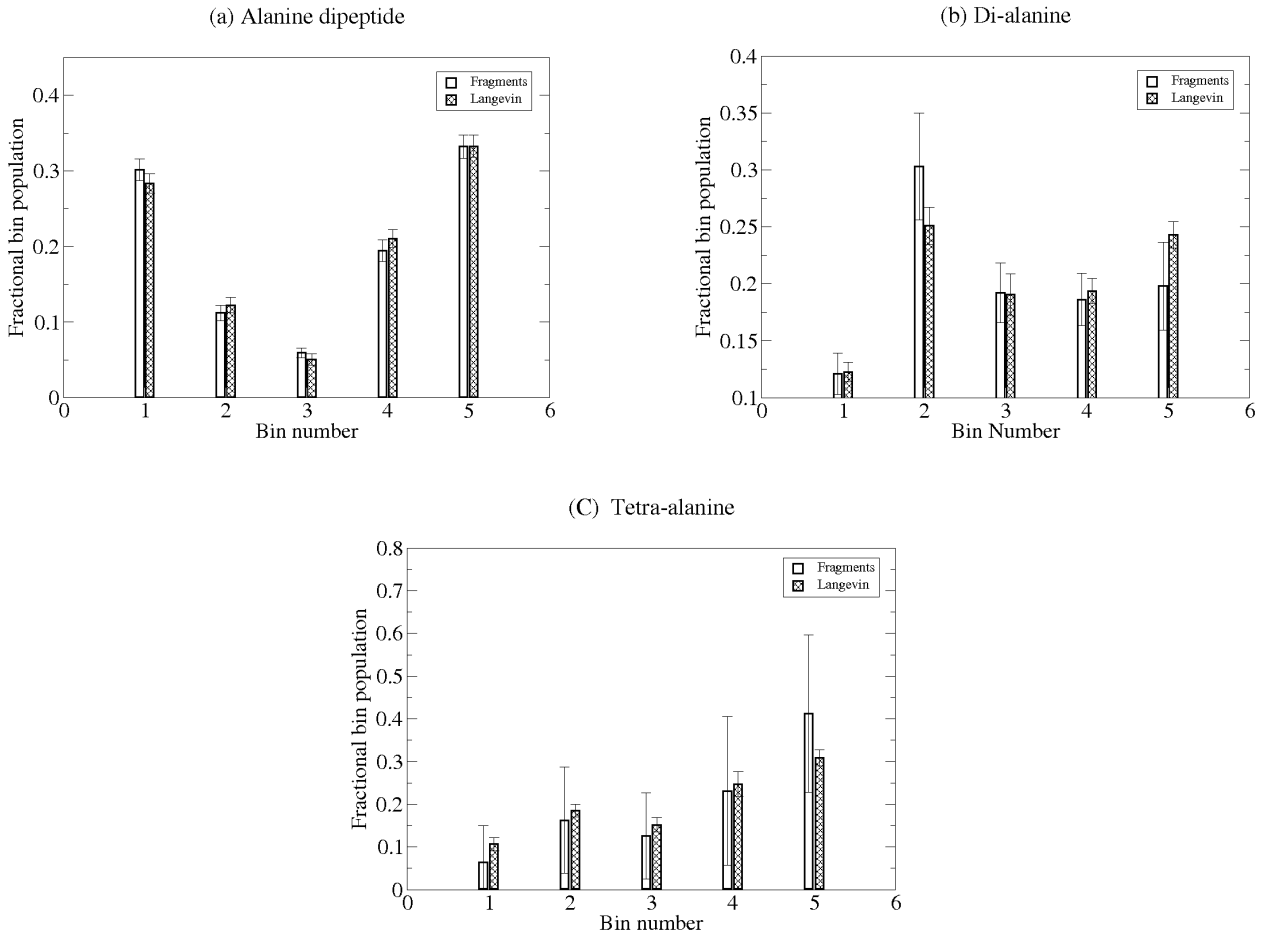

Figure 10: Comparison of equilibrium distributions from fragment combination and Langevin simulation.The graphs show the fractional population in different regions of configuration space, as described in Sec.II K. Three peptides are considered: (a) alanine dipeptide, (b) di-alanine, and (c) tetra-alanine. The error bars for both the fragment combination and Langevin results reflect twice the standard deviations among 20 independent simulations, roughly a 95% confidence interval. Each Langevin simulation was 50 nsec long. The statistical agreement is good in every case.

Free energy terms for di-alanine from Eq. (III.21) and for tetra-alanine from Eq. (III.22)			
Di-alanine		Tetra-alanine	
Term	Estimate [kcal/mol]	Term	Estimate [kcal/mol]
F_{Ace}	14.783(0.003)	F_{Ace}	14.783(0.003)
F_{Ala}	33.326(0.015)	F_{Ala}	33.326(0.015)
F_{Nme}	16.574(0.003)	F_{Nme}	16.574(0.003)
$\Delta F_{\text{Ace}\to\text{Ala}}$	−0.801(0.002)	$\Delta F_{\text{Ace}\to\text{Ala1}}$	−0.801(0.002)
$\Delta F_{\text{Ala}\to\text{Ala}}$	−0.771(0.014)	$\Delta F_{\text{Ala1}\to\text{Ala2}}$	−0.774(0.013)
$\Delta F_{\text{Ala}\to\text{Nme}}$	−0.499(0.013)	$\Delta F_{\text{Ala2}\to\text{Ala3}}$	−0.774(0.012)
$\Delta F_{\text{nonbonded}}$	−0.809(0.031)	$\Delta F_{\text{Ala3}\to\text{Ala4}}$	−0.771(0.014)
		$\Delta F_{\text{Ala4}\to\text{Nme}}$	−0.498(0.009)
		$\Delta F_{\text{nonbonded}}$	−1.986(0.284)
F^{phys}	95.128(0.057)	F^{phys}	159.057(0.293)

Table 4: Free energy terms used in calculating the absolute free energy for di-alanine and tetra-alanine. The table gives free energy values in kcal/mole, as well as two standard deviations (in parentheses) based on 20 independent calculations.

IV. AUTOMATED SAMPLING EFFICIENCY ASSESSMENT

A. OVERVIEW

I introduce the growth algorithm in Chapter II and discuss how to calculate absolute free energy by staging the free energy difference calculation in Chapter III. In this chapter I will address the question "Is the growth algorithm faster and more efficient than standard Langevin? How much faster?" In order to address this question, one needs a way to find the efficiency quantitatively. As was mentioned in Chapter I, one needs to estimate the effective sample size (ESS) of trajectories generated by different type of algorithms– dynamical or non-dynamical. Let us start with the importance of efficiency assessment. This chapter has been adapted from a published paper [119].

1. The importance of efficiency assessment

The field of molecular simulations has expanded rapidly in the last two decades and continues to do so with progressively faster computers. Furthermore, significant effort has been devoted to the development of more sophisticated algorithms[30, 86, 120–122] and forcefields [13–15, 123, 124] for use in both physical and biological sciences. To quantify progress – and indeed to be sure progress is occurring – it is critical to assess the efficiency of the algorithms. Moreover, if the quality of sampling is unknown, we cannot expect to appreciate fully the predictions of molecular mechanics forcefields: after all, statistical ensembles, whether equilibrium or dynamical, are the essential output of forcefields. These issues demand a gauge to assess the quality of the generated ensembles[125] – one which is automated, non–subjective, and applicable regardless of the method used to generate the ensembles.

Ensembles are of fundamental importance in the statistical mechanical description of physical systems: beyond the description of fluctuations intrinsic to the ensembles, all thermodynamic properties are obtained from them. [126] The quality of simulated ensembles is governed by the amount of "information" present in the ensemble. Due to significant correlations between successive frames in, say, a dynamics trajectory, the amount of information cannot be directly gauged from the total number of "frames". Rather, the number of statistically independent configurations in the ensemble (or the effective sample size, ESS) is required.[26, 127–129] This effective sample size has remained difficult to assess for reasons described below. In this work, we present a straightforward method to determine the ESS of an ensemble – regardless of the method used to generate the ensemble – by quantifying variances in populations of physical states.

2. Historical background of sample size calculation

A conventional view of sample size based on a dynamical simulation is given by the following equation:

$$\mathrm{ESS} = \frac{t_{\mathrm{sim}}}{t_{\mathrm{corr}}[f]} \tag{IV.1}$$

where t_{sim} is the simulation time, and $t_{\mathrm{corr}[f]}$ is the correlation time[26] for the observable f, which is presumed to relax most slowly. However, the estimation of the correlation time is data intensive and potentially very sensitive to noise in the tail of the correlation function. [23] Other approaches for assessing correlations have, therefore, been proposed. For example, Mountain and Thirumalai[130, 131] introduced the "ergodic measure", which quantifies the time required for the observable to appear ergodic. Flyvbjerg and Petersen [23] developed a blocking averaging method that can be adapted to yield a correlation time and ESS.[35]

The key challenge in applying Eq. (IV.1), however, is the choice of an observable f which consistently embodies the slowest motions across the incredible variety of molecular systems. Indeed, it is well appreciated that different observable exhibit different correlation times. (*e.g.,* Ref. [29]) For example, in a typical molecule, bond lengths become decorrelated faster than dihedral angles. Nevertheless, apparently fast observable rarely are fully decoupled from

the rest of the system: slower motions ultimately couple to the fast motions and influence their distributions in typical cases.[29] On the whole, there is significant ambiguity in the use of a hand–picked observable to estimate "the" correlation time – not to mention, subjectivity. Moreover, the ultimate goal of simulation, arguably, is not to compute a particular ensemble average but to generate a truly representative ensemble of configurations, from which any observable can be averaged.

Several years ago, Lyman and Zuckerman proposed that the configuration–space distribution itself could be used as a fundamental observable.[25] In particular, it was pointed out that if configuration space is divided into different regions or bins, then the resulting "structural histogram" of bin populations could be a critical tool in assessing sampling. The idea was subsequently used to quantify sample size in at least two studies: Lyman and Zuckerman developed a scheme to quantify ESS for trajectories with purely sequential correlations based on variances in the bins of the structural histogram;[26] Grossfield and coworkers suggested a bootstrapping approach for estimating ESS based on structural histograms.[129] The present work expands on ideas from these studies.

This study extends the earlier structural–histogram approaches by focusing on "physical states". Qualitatively, a physical state can be defined as a region of configuration space for which the internal timescales are much shorter than those for transitions between different physical states.[132] The populations of physical states seem an intuitive choice for quantifying sampling quality, since they reflect slow timescales by construction. Indeed, the state populations along with state definitions (addressed in Sec. IV.B.1) can be said to embody the equilibrium ensemble. This type of argument can be made semi–quantitative by noting that any ensemble average $\langle f \rangle$ can be expressed in terms of state populations p_i and state-specific averages $\langle f \rangle_i$ for state i, because $\langle f \rangle \simeq \sum_i p_i \langle f \rangle_i$. Thus, the goal of sampling can be described as obtaining both (i) state populations and (ii) well-sampled ensembles within each state.

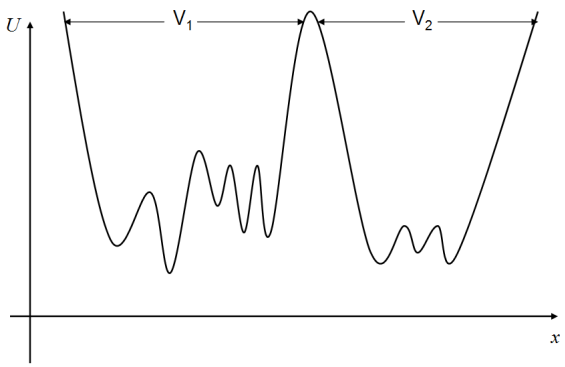

Figure 11: A schematic two-state potential energy landscape from Eq. (IV.2). The states are defined by the "volumes" V_1 and V_2. The distributions of configurations within each states help to determine the overall ratio of state populations in Eq. (IV.2).

Statistical mechanics principles strongly suggest, moreover, that state populations should be viewed as the slowest converged observables. To see why, consider states i and j defined by regions of configuration space V_i and V_j. The ratio of state populations is given by the ratio of state partition functions:

$$\frac{\text{prob}(i)}{\text{prob}(j)} = \frac{Z_i}{Z_j} = \frac{\int_{V_i} d\mathbf{r} \exp(-U(\mathbf{r})/k_\text{B}T)}{\int_{V_j} d\mathbf{r} \exp(-U(\mathbf{r})/k_\text{B}T)} \tag{IV.2}$$

where Z_i is the partition function for state i, U is the potential energy of the system, T is the temperature, and $\mathbf{r}$ represents all configuration–space coordinates. Eq. (IV.2) indicates that state populations cannot be determined without good sampling within each state. In other words, it would seem impossible for an algorithm (which is correct for arbitrary systems) to predict state populations without having already sampled correctly within states (see Fig. 11). For this reason, the state populations can be considered the fundamental set of slow observables – a physically motivated choice of structural histogram. We will use variances in state populations to estimate ESS, an approach which applies to both dynamic and non-dynamic (*e.g.*, exchange) simulations.

Accordingly, an important prerequisite for the estimation of ESS is the determination of physical states. In this work, we use a particularly simple method for the approximation of physical states that uses information present in a dynamics trajectory regarding the transition rates between different regions. Regions showing high transition rates with each other are assumed to belong in the same physical state. Further, this procedure also highlights the hierarchical nature of the energy landscape. The state approximation scheme is based on ideas of Chodera *et al.*[132] who developed approximated metastable states by determining a division of the total configuration space that maximizes the self transition probabilities (*i.e.*, the divisions represent metastable states.) See also Ref. [133]. The state–approximation method can also be used with short dynamics trajectories initiated from configurations obtained from non–dynamic simulations.

We emphasize , nevertheless, that this procedure for ESS estimation can be used with states discovered by different means.

The discussion of this project is organized as follows. In this chapter, I first describe in detail the procedure we use to estimate the effective sample size. Then, I present results for several models with different levels of complexity – a two–state toy model, butane, calmodulin, di–leucine, and Met–enkaphalin. The ESS results are compared with the previous "decorrelation time" approach [89]. We also analyzed multi-μsec atomistic simulations for the membrane protein rhodopsin. [129] I then discuss the practical aspects of the procedure and present conclusions. Further, in Chapter V, I describe the simple, automated procedure used to determine approximate physical states.

As for the systems analyzed here, met-enkaphalin and di-leucine trajectories were generated by Dr.Edward Lyman, Dr.Bin W. Zhang simulated the calmodulin, and I generated butane trajectories. Plus, I analyzed Dr.Alan Grossfield's rhodopsin trajectories as a coarsed-grained case. All the rest of the work analyzing the trajectories was performed by the author. Dr.Divesh Bhatt helped edit the draft of a manuscript for publication.

B. METHODS AND SYSTEMS

We have argued above that the populations of physical states are fundamental observables for assaying the equilibrium ensemble. We therefore propose that the statistical quality of an equilibrium ensemble be quantified using variances in state populations. As usual, the variances will decrease with better sampling. Importantly, however, simple binomial statistics permit a fairly precise quantification of the ESS – *i.e.*, the number of statistically independent configurations to which an ensemble is equivalent – regardless of the number of configurations in the original ensemble. Below, I will address the issues of computing variances from dynamical and non–dynamical simulations, as well as methods for approximating physical states.

The key technical idea in connecting the variance in a state's population to the ESS follows work presented in Ref. [26]: an analytic form for the variance can be computed based on a known number of independent samples. If one "turns around" this idea, given the observed variance, an estimate for the number of independent samples can be immediately obtained. In particular, given a region j of configuration space with fractional population p_j, the variance in p_j based on N independent samples is $\sigma_j^2 = p_j(1-p_j)/N$. (In practice, this variance is estimated from repeated independent simulations, each yielding a value for p_j.) The ESS based on populations recorded for region j can therefore be estimated via

$$N_j^{\text{eff}} = \frac{\bar{p_j}(1-\bar{p_j})}{\sigma_j^2} \tag{IV.3}$$

where $\bar{p_j}$ is the observed average population in region j. As noted in Ref. [26], Eq. (IV.3) is actually a limiting form appropriate for large N. Although it is straightforward to include corrections accounting for the fact that only $N-1$ observations are independent (because $\bar{p_j}$ is the observed average among the p_j values used in estimating the variance), the effect is unimportant compared to the intrinsic fluctuations in N^{eff}.

Each region or state will yield its own estimate for the ESS via Eq. (IV.3), but we are interested in the smallest ESS reflecting the slowest timescales. As described below, in this thesis, a hierarchical decomposition of configuration space is used, which leads to only two states at the top level, by construction. In turn, these two states yield identical ESS values

by Eq. (IV.3). Alternatively, if a full hierarchy is not constructed, one can simply select the lowest ESS value as the best quantification of sampling, reflecting the worst bottleneck encountered.

Noted that, the minimal value which can be determined is one, and generally $N_j^{\text{eff}} > 1$. It is because the minimal number of independent configurations is one. Thus, a value of less than 10, is strongly suggestive of inadequate sampling.

1. Hierarchical approximation of physical states

The approximation of physical states has previously been addressed in some detail, particularly in the context of developing Markov models.[132] In this chapter and in Chapter V, I describe a simpler approach used in this work. As we elaborate in the Sec. IV.D, it appears that the ESS analysis does not require a particularly precise specification of physical states. Because the prescription is to find the slowest timescale (*i.e.*, smallest ESS) among the many which may be present, and because the physical states are reasonable, the approach works reliably. On the other hand, although Eq. (IV.3) can be applied to an arbitrary region in principle, it can "get fooled" into over–estimating the ESS if only a small part of a state is considered: see Sec. IV.D for details.

We emphasize that the ESS analysis described above is distinct from the states analyzed, and other reasonable state decomposition procedures can be used.

Chapter V details the hierarchical state approximation scheme adopted here, which is closely related to the work of Chodera *et al.*[132] In brief, given the best data available, we first divide configuration space into small regions or bins (following Refs. [26] and [69]), which do not necessarily correspond to energy basins. Based on one or more dynamical trajectories (perhaps those being analyzed for ESS), we estimate transition rates among each pair of regions. Starting from the fastest pairwise rates, the bins are combined into state–like aggregated regions. By construction, all pairwise rates within each aggregate are faster than rates between aggregates. The process is continued to construct a full hierarchy until all aggregates are combined (see Figures 13 and 14 in Chapter V). The approximate states used to estimate the ESS are based on the top (i.e., two–state) level of the hierarchy,

which reflects the slowest timescales as desired.

The rate estimation procedure is well–suited to the purpose of ESS estimation. First, it is fairly simple and typically requires a small fraction of the computational cost of the simulation being analyzed. More importantly, as noted in the Sec. IV.D, it performs as well as a somewhat more complex approach we implemented (data not shown). Although the procedure (and others[132]) requires dynamical trajectories to estimate inter–bin transition rates, this does not mean prohibitively expensive dynamics simulations must be performed, as we now discuss.

a. State approximation from non–continuous dynamical trajectories Because the state approximation scheme depends on continuous dynamical trajectories, the question arises as to how states can be obtained when sampling has been performed using a non–dynamical method such as replica exchange.[30, 57, 134] Although exchange simulations use continuous trajectories which contain the necessary information for estimating rates among local regions,[135] other sampling methods may not employ dynamical trajectories at all (*e.g.*, see Ref. [69]).

In fact, states can be approximated based on a set of short dynamics trajectories run after a possibly costly non–dynamical trajectory. In particular, a set of M trajectories (we use $M = 20$ below) can be initiated from random configurations selected from the best available simulation. These short trajectories need only be long enough to permit exploration *within* states. There is no need for transitions between states. The only modification to the state approximation scheme described previously is that it may not be possible to iterate the combination procedure until all states are combined. Rather, the process will terminate after regions with measurable transition rates are combined. A set of approximate states will remain for which no inter–state transitions have been recorded. For each of these remaining states, an ESS estimate can be obtained via Eq. (IV.3). Because of the interest in the slowest timescales, the overall ESS will be taken as the minimum among the various state values.

The scheme just described is tested below, and compared with the use of longer trajectories for state approximation.

2. A caveat: Self–consistent but not absolute ESS

Without prior knowledge or assumptions about a landscape, it would appear impossible to know whether every important state has been visited in a given simulation. This is not a limitation of the analysis per se, but of any attempt to estimate ESS based on simulation data. Nevertheless, it is important to make this caveat clear.

Therefore, the goal of the present analysis is not to assess the coverage of configuration space, but to self–consistently assess sampling quality given the states visited in the simulation. In other words, we answer, "What is the statistical quality of the sampling based on the configurational states visited in a given set of simulations?" The ESS estimation can therefore be viewed as an upper bound to the true ESS based on the full configuration space. ESS estimation, nevertheless, is essential for assessing efficiency in algorithms and precisely specifying the predictions of modern forcefields.

On the other hand, so long as a state has been visited in a simulation, it can greatly affect the sample size. For instance, if a state has been visited only once among multiple independent simulations, the estimate of its population variance will be large and lead (correctly) to a small ESS.

3. Estimating variances in state populations

The heart of this approach is to estimate ESS based on variances in state populations using Eq. (IV.3). Clearly, then, without reliable variance estimates, we cannot expect ESS values to be reliable.

For dynamical simulations - i.e., simulations yielding trajectories in which correlations are purely sequential, such as MD and "ordinary" (Markov chain) MC - there is more than one way to estimate a variance suitable for ESS calculation via Eq. (IV.3). Ideally, a number of independent dynamics runs would be started from significantly different initial conditions. Nevertheless, multiple simulations started from the same configuration will also reveal the variance associated with the duration of each run: for instance, if only one simulation makes a transition to an alternative basin, a large variance and small ESS estimate will result, appropriately. It is important to note that the ESS thus calculated is characteristic of one of

the trajectories, so that M independent trajectories imply an ESS which is M times as large. This discussion also indicates that a single long trajectory can be divided into M segments ("Flyvbjerg1989s") which can be used for variance estimation.

More complex simulation methods, such as replica exchange,[30, 57, 134] may require multiple independent runs for careful variance estimation. To see why in the case of replica exchange, note that continuous trajectories will traverse a ladder of different "conditions" (*e.g.*, temperatures or forcefields), but often only a single condition is of interest. By the construction of such an algorithm, configurations appearing at one time at the condition of interest may be strongly correlated with configurations occurring later on - but not with configurations in between, when a different trajectory may have occupied the condition of interest. In sharp contrast to dynamics simulation, correlations may be non–sequential. This absolutely precludes estimating the variance by simply cutting up the equilibrium ensemble into Flyvbjerg1989s or segments. Such a variance may not reflect sampling quality, and could misleadingly reflect only diffusivity among ladder levels.[29]

For a non-dynamical simulation method, the only sure way to estimate a variance which reflects the underlying ESS is by multiple independent runs. The extra cost could be modest if each run is sufficiently short and such runs would, of course, enhance sampling.–*i.e.,* they would "pay for themselves." In any case, the cost seems worthwhile when it permits careful quantification of the results. We note that subtleties in estimating uncertainties in replica exchange simulations have been noted previously.[136–138]

4. Systems studied

We study several systems using the ESS procedure described above to establish correctness and robustness of the procedure. The systems range from toy models and small molecules to coarse-grained and atomistic proteins.

a. Toy models with known sample size First, we study simple toy models for which the correct sample size is known in advance, to establish the correctness of the procedure. The toy system has n idealized "states" that correspond to pre-set values of independently

drawn random numbers. The sample size in such toy models is simply the number of random numbers drawn by construction. We use two such toy models: $n = 2$ (and both states with equal population), and $n = 5$ (with state probabilities 0.1, 0.15, 0.2, 0.25, 0.3). In practice, a random number in the range of [0,1] is picked and assigned to specific state. The number of random numbers is the true effective sample size since these random numbers are totally independent. An application of Eq. (IV.3) to the two–state system yields, by construction, the same sample size in both the states. On the other hand, the effective sample sizes obtained may, in general, be different when the number of states is greater than 2. Thus, the five–state toy model is useful in determining the consistency in the sample sizes obtained in the different states.

The sampling in these toy models is nondynamic and uncorrelated. Thus, the use of such models illustrate the applicability of the effective sample size determined by Eq. (IV.3) to nondynamic sampling. Results for the toy models and all other systems are given in Sec. IV.C.

b. Systems with *a priori* known physical states In contrast to independent sampling in the toy models, dynamics-based sampling in molecular systems is not typically independent and the sample size is not known in advance. Nevertheless, a knowledge of physical states allows for an independent estimate of the ESS by computing the variances in the known physical states and comparing with the estimate obtained via approximate hierarchical states. Thus, the robustness of the procedure described in Sec. IV.B with regard to definitions of physical states can be checked. We study two such systems with *a priori* known states: butane and calmodulin. A second, independent ESS estimate for these systems is derived from a time correlation analysis [130].

I have studied a standard all–atom butane model using the OPLSAA forcefield [15]. This system has three well–known states: trans, gauche+, and gauche-. The 1 μsec dynamical trajectory is generated at 298K using Langevin dynamics (as implemented in Tinker v. 4.2.2) in vacuum with friction constant 91/ps.

I also study the N-terminal domain of calmodulin, which has the two known physical states: the apo form (PDB id – 1CFD) and the holo form (PDB id – 1CLL). A long trajectory

($5.5 * 10^7$ MC sweeps) was generated by using "dynamic" Monte Carlo (small, single-atom moves only) as previously described [139]. To permit transitions, we use a simple alpha–carbon model with a double–Gō potential to stabilize the two physical states. Full details of this model are given elsewhere.[139]

c. Systems with unknown physical states For most biomolecular systems, the physical states are not known in advance. For this reason, I have tested the method on several such systems, starting with two peptides: leucine dipeptide (acetaldehyde–$(\text{leucine})_2$–n–methylamide) and Met–enkaphalin (NH_3^+–Tyr–$[\text{Gly}]_2$–Phe–Met–COO^-). I use the Charmm27 forcefield for leucine dipeptide and OPLSAA forcefield for Met-enkaphalin and generate trajectories using overdamped Langevin dynamics (in Tinker v 4.2.2) at 298 K with a friction constant of 5/ps for both. For leucine dipeptide I use a uniform dielectric of 60, and the GB/SA solvation for Met-enkaphalin. [140, 141] For each system, a 1 μs simulation is performed with frames stored every 1 ps for Met-enkaphalin and every 10 ps for leucine dipeptide.

I then study a much more complex system – rhodopsin [129, 142]. I analyze 26 independent 100 ns molecular dynamics simulations of rhodopsin in a membrane containing 50 1–stearoyl–2-docosahexaenoyl–phosphatidylethanolamine (SDPE) molecules, 49 1–stearoyl–2-docosahexaenoyl–phosphatidylcholine (SDPC) molecules, and 24 cholesterols. There is an explicit water environment embedded in a periodic box. The all-atom CHARMM27 forcefield was used. We analyze only protein coordinates under the assumption that these will include the slowest timescales.

5. Independent ESS estimates

We would like to compare ESS estimates obtained from the new procedure to independent "reference" results. Independent ESS estimates can be obtained in several ways, depending on the system and simulation method to be analyzed.

For uncorrelated sampling in the toy models, the ESS is known in advance: it is simply the number of samples used to obtain the state variance. In this case, we merely check that

knowledge of the variances along is sufficient to recover the number of samples.

In some molecular systems, such as butane and calmodulin in this study, physical states are known in advance. Independent variance (and hence ESS) estimates are then obtained using these as "exact states". These are compared to ESS estimates obtained fully automatically based on states approximated from trajectories. In systems with a small number of states, additional ESS estimates can be approximately obtained simply by counting transitions.

Whether or not physical states are known, if a dynamics (or Markov Chain MC) trajectory is analyzed, independent ESS estimates can be obtained using the previously developed structural decorrelation time analysis [26] and Eq. (IV.1). This approach uses a t_{corr} reflecting the time to sample the whole distribution. In work with model one-dimensional systems (data not shown), Lyman and Zuckerman have found that the ESS is estimated within a factor of 2 using the method of Ref. [26]; therefore ESS estimates based on decorrelation time are shown as ranges.

C. RESULTS

1. Non–dynamic toy systems

First, we establish the formal correctness of the method for estimating N^{eff}. For this purpose, we study the toy models described in Sec. IV.B.3 for which the sample size is known in advance. For each toy model, we draw N independent samples and estimate the sample size using the procedure described in Sec. IV.B.

To determine whether an accurate estimate of N^{eff} ($\equiv N$) is obtained, we also compute both the mean value and standard deviation of N^{eff}. As suggested by Eq. (IV.3), this requires computation of variances of both the mean population and the population variance (these quantities are equal across the states for a two–state system). Further, care must be taken to account for the nonlinear dependence of N^{eff} on the state variance in Eq. (IV.3).

For the two–state model, with $N = 2000$, we obtain a mean value of $\langle N^{\mathrm{eff}} \rangle = 2004$, with

	approx. states			known states	time correlation	counting
	1	2	3			
butane	6064	6236	6200	5865	5000–10000	6000
calmodulin	93	90	92	91	80–160	80

Table 5: Automated and independent effective sample sizes for butane and calmodulin. ESS estimates obtained from Eq. (IV.3) using three different sets of approximate physical sets are shown in Columns 2–4. Also shown are ESS estimates from Eq.3 and the known physical states (column 5), the structural decorrelation time analysis (column 6) and from counting the number of transitions (column 7).

a standard deviation of 57.4. Similarly, for $N = 4000$, we obtain a mean $\langle N^{\text{eff}} \rangle = 4041$ with a standard deviation 117.6. This confirms the basic premise of using Eq. (IV.3) based on the binomial distribution. The intrinsic fluctuations in the estimates, about 3% in both cases, presumably do not decrease with increasing N due to the non–linearity of Eq. (IV.3).

In the five–state model estimates of the sample sizes in each state are different (see Sec. IV.B), and such a model is a further step in confirming Eq. (IV.3) in a mere heterogeneous case. Using $N = 2000$, and states with fractional populations 0.1, 0.15, 0.2, 0.25, and 0.3, the mean sample sizes (standard deviations) are obtained as 2007 (70), 1998 (57), 1974 (35), 1966 (79), and 1986 (63), respectively. There is a good agreement across the states, as well as with the correct sample size $N = 2000$.

2. Systems with *a priori* known physical states

We turn next to molecular systems with known physical states for which long dynamics trajectories are available. This is essentially the simplest case for a molecular system, because two independent estimates of ESS can be obtained, as described below. Comparison of this blind, automated procedure to these independent estimates further establishes the correctness and robustness of the procedure. Additionally, because this automated state-

	approx. states			time correlation
	1	2	3	
di–leucine	1982	1878	1904	1100-2200
Met–enkaphalin	416	362	365	250–500

Table 6: Effective sample sizes for di–leucine and Met–enkaphalin. Eq. (IV.3) is used on the final two states in the hierarchical picture obtained by three different repetitions of the binning procedure (Columns 2–4), and the ESS is independently estimated from the structural decorrelation time correlation (Column 5).

construction procedure is somewhat stochastic (see Chapter V), we repeat the procedure to understand the fluctuations in the ESS estimates.

We obtained multiple estimates of ESS as described above using a single long trajectory for each of the two systems with known physical states – butane and calmodulin. Table 5 shows results for N^{eff} for the two systems, including three different estimates of N^{eff} from Eq. (IV.3) based on different sets of approximate states. Comparison is also made to the use of Eq. (IV.3) based on known physical states, and to the range of effective sample sizes obtained using time correlation analysis. For both butane and calmodulin, the procedure is very "robust" in estimating N^{eff}, as different binning procedures give similar estimates. These estimates also agree with the range of sample sizes suggested by the correlation time analysis and with counts of transitions. For butane, the total number of transitions among the three state is about 6000. For calmodulin, the total number of transitions is 80. These results also agree with the estimates in Table 5.

3. Systems with unknown physical states

Exact physical states are not known in advance for most biomolecular systems. Thus, we test the approach described in Sec. IV.B to determine ESS in three such systems - dileucine, Met–enkaphalin and rhodopsin. Because the physical states are not well defined, we can only

obtain independent estimates from the time correlation analysis. A single 1 μsec trajectory is analyzed for each of the peptides, whereas 26 trajectories of 100 nsec each are studied for rhodopsin.

Table 6 shows repeated ESS estimates using the approximate states with Eq. (IV.3) as well as the time–correlation analysis for both dileucine and Met–enkaphalin. There is good agreement between the variance-based estimates and those from time correlation analysis for both systems.

We proceed to analyze the sample size of 26 rhodopsin trajectories based on the approximate states with Eq. (IV.3). The analysis gives three physical states, with sample sizes 1.93, 1.99, 2.73, respectively, per 100 nsec trajectory. The three states are never further connected in full hierarchy, since transitions are not observed between some bin pairs. The three N^{eff} estimates, nevertheless, are quite similar and all are less than 10. However, Eq. (IV.3) always yields a value ≥ 1, indicating that the 100 nsec rhodopsin values are effectively minimal and reflect inadequate sampling. In Ref. [129], Grossfield and coworkers examined the same trajectories with principal components and cluster populations. They concluded, similarly, that rhodopsin's fluctuations are not well described by 100 ns of dynamics, and that the sampling is not fully converged even for individual loops.

4. Application to discontinuous trajectories

Although sample size estimation using Eq. (IV.3) is applicable to non–dynamical simulation methods, the underlying physical states, (approximated from transition rates between regions of configuration space: see Chapter V), may not be easy to calculate from non–dynamical trajectories. We therefore investigate the feasibility of running short dynamics trajectories starting from configurations previously obtained from non–dynamic simulations and then estimating ESS based on states from the short dynamics simulations.

For this purpose, we ran a series of 20 short Langevin simulations for both di-leucine and Met-enkaphalin, starting from configurations obtained in the original long trajectories (which serve as proxies for well-sampled ensembles by an arbitrary method). For both systems, we approximated states as described in Sec. IV.B.1, and estimated the ESS as $\min_j\{N_j^{\text{eff}}\}$. For

simulation segments as short as 200 psec we could obtain the correct ESS within a factor of 2 (di-leucine) or 3 (Met-enkaphalin), whereas the longest timescales in these systems exceed a nsec. [26] However, a precise estimate of the ESS required 1-3 nsec segments.

Note that Chodera *et al* [132] also used discontinuous trajectories in their state approximation scheme. As noted in the Chapter V, this scheme is a simplified version of theirs.

5. Spurious results from un–physical states

Thus far, we have focused on using physical states with Eq. (IV.3), based on the arguments presented in the Sec. IV.A. In principle, however, Eq. (IV.3) can be applied to an arbitrary region. To confirm the need for using physical states, here we investigate what happens when only part of a state is used. We will see that spurious ESS estimates results.

The system we examine is butane. We divide the configuration space into 10 "bins" using Voronoi cells [77], and perform *no* combination into physical states. We estimate the effective sample size using Eq. (IV.3) for each bin. We examine a 1 μ sec trajectory, for which $N^{\text{eff}} \simeq 6000$.

Table 7 shows estimates of ESS obtained for each of the 10 arbitrary bins, which are not approximate states. The estimates shows a dramatic bin dependence.

The problem with using bins rather than states results for simulations which use dynamics. In fact, arbitrary bins can be used in Eq. (IV.3) if sampling is fully uncorrelated; we verified this using a fixed number of butane configurations which were essentially uncorrelated. However, when dynamics are present, the variance of one bin is a convolution of state variances and fast processes. I will discuss this in more detail in Sec. IV.D.

D. DISCUSSION

1. Diagnosing poor sampling

A key outstanding issue is how to know when sampling is inadequate, at least in the self-consistent sense of Sec. IV.B.2. The "diagnosis" of poor sampling is intimately connected

Bin number	ESS
1	12567
2	61391
3	82087
4	91839
5	292655
6	71194
7	240201
8	5600
9	162731
10	210261

Table 7: Spurious ESS estimates when physical states are not used. Butane sample size is estimated in each of 10 arbitrary regions of configuration space. The actual sample size is $\sim$6000, based on a 1 μsec Langevin dynamics trajectory.

with the idea of estimating ESS by subdiving a dynamics trajectory into smaller, equal segments.

First, consider subdividing a dynamics trajectory into smaller, equal segments to estimate the population mean and variances. If the trajectory is very long compared to all corelation times, no serious problems will arise. If the sample size estimate for each of these segments is less than 10, however then the method does not reliably give the estimate of the sample size of the total trajectory, and likely overestimates it. For example, if the correct total number of independent configurations in the full trajectory is 10, and we subdivide it into 20 equal segments, then each of the segment will give a sample size of 1, which is the minimum number possible using Eq. (IV.3). This leads to an overestimate of the sample size. But the problem is easily diagnosed by ESS $\sim$ 1 for each segment. If division into fewer segments still leads to ESS $\sim$ 1, sampling is likely inadequate.

2. The inadequacy of arbitrary regions for ESS estimation

It is somewhat difficult to understand the reason for spurious results for ESS obtained using a correlated dynamics trajectory from bins that are a small part of a physical state as in Sec. IV.C.5. A two-state thought experiment is instructive. Consider a system with two basins, A and B, separated by a barrier. Imagine that we divide the full space into many bins, of which the seventh is a small part of state A and has the (true) probability of p_7. In ideal uncorrelated sampling, the observed outcomes should be in the bin with probability p_7 and out of the bin with probability 1-p_7. However, in dynamical sampling, if the system is trapped in state A (with a fractional population p_A) for the observation time, the observed probability in the bin turns out to be p_7/p_A instead of p_7. Conversely, if a trajectory segment is trapped in state B, the observed population of bin 7 is zero. The variance of this observed distribution when $p_7 \ll p_A$ is much lower than the binomial case; physically, the fast timescales within state A act to "smooth out" population variation within a small part of the state. The estimated ESS obtained using a correlated (*i.e.*,dynamical) trajectory will appear to be larger based on such a bin, as occurs in Table 7.

V. DISCOVERY OF PHYSICAL STATES IN A HIERARCHICAL PICTURE

A. REVIEW

In the previous chapter, I demonstrated how to estimate the ESS based on the variance in populations of physical states. The key step of this method is to find the physical states. Physical states of a system should be described in a hierarchical way. The hierarchical description of physical states was initiated by the classical studies of Wolynes and collaborators on the reaction kinetics of myoglobin and oxygen [143]. There is also study about protein folding by using hierarchical description of energy landscape [144]. Hierarchical disconnectivity graphs [28] have played a key role in recent efforts to explain how diverse processes such as protein folding, crystallization and self-assembly. Disconnectivity graphs are schematic descriptions of the energy landscape and provide the relation among local energy minima. Since the number of local minima increase exponentially with the number of atoms [28], the graph could be really messy and miss the big picture of the free energy landscape of the system. We developed a method to describe the systems of interest in large "clusters" (physical states) and give the hierarchical picture of the physical states based on the transition rates between neighbouring states. In the ESS estimation, we are only interested in the final stage – two states, which are separated by high energy barrier. The benefit of discovery of physical states is beyond ESS estimation: it is very important for understanding dynamics of biomoelcular processes. Note that a physical state in this thesis is a region of configuration space that contains many configurations. They are separated by high energy barriers.

Almost all the biomolecular processes are fundamentally dynamic in nature. In many complex systems of physical importance, transitions take place between stable states sep-

arated by a high (free) energy barrier. Examples are isomerizations in clusters, chemical reactions, protein folding and crystal nucleation. Molecular simulation techniques such as molecular dynamics (MD) in principle enable the computation of the reaction rate constants, the search for transition states and the exploration of reaction mechanisms. But since the rate constant of the transition depends exponentially on the activation barrier height [28], the expectation time of a transition can easily become orders of magnitude longer than the molecular timescale which is usually measured in femtoseconds. A purely static description of these motions is not enough for mechanistic "understanding" the dynamical nature of these processes.

To study the kinetic pathways of the systems, it is necessary to decompose the conformational space into a set of physical states. There is study of automated discovery of metastable states for the construction of Markov models of dynamical simulation by Chodera and coworkers [132]. It shows excellent results but the drawbacks are that it is complicated and requires the user to choose the number of states in advance. In this study, we extend their study and develop a simpler and faster method. Also we borrow the idea to investigate the physical states at different hierarchical levels [143].

In this Chapter, I describe the physical state discovery method and its results. In this method, bins or regions in configurational space are combined to give the physical states, as discussed below in more detail. This method is based on the work of Chodera *et al.*[132], but is simpler. There is no Markovian requirement on the selection of bins. Indeed, a typical bin in a configurational space for a large multidimensional system may itself encompass several separate minima. We emphasize that the procedure is designed solely for the purpose of estimating sample size and is not claimed to be an extremely precise description of states.

The approach explicitly shows the hierarchical nature of the configurational space [145, 146], and ultimately focuses on the slowest timescale – which is of paramount importance for the estimation of the effective sample size.

The trajectories analyzed in this chapter are the same as previous chapter, some of which were generated by others. I performed the analysis of the trajectories, some of which are simulated by Dr.Divesh Bhatt who also provides important insights.

B. METHODS

1. Use of rates to describe conformational dynamics

The approximate states are constructed based on rates between regions of configuration space, which are a fundamental property that emerges uniquely from the natural system dynamics. Following Ref. [132], we first decompose the conformational space into multiple bins as detailed below. Subsequently, we combine bins that have the highest transition rates between them, iterating to create a hierarchical description. This procedure is based on the physical idea of separation of time scales: there are faster timescales (high transition rates) associated with regions within a single physical state, and slower timescales for transitions between states. Furthermore, "fast" and "slow" timescales are not absolute, necessitating a hierarchical description following precedents.[145, 146]

2. Binning decomposition of the configurational space

We divide the whole configuration space into m bins, and determine the physical states by combination of these regions. All data reported here used $m = 20$. The procedure to decompose the whole configurational space (with N configurations) into m bins is as follows:[69]

I. A reference configuration i is picked at random from the trajectory.

II. The distance of the configuration i to to all remaining configurations in the trajectory is then computed, based on an appropriate metric discussed later.

III. The configurations are sorted according to distance, and the closest N/m configurations are removed.

IV. Steps 1–3 are repeated $m-1$ times on the progressively smaller set of remaining configurations, resulting in a total of m reference configurations.

For the distance metric, we select the root-mean squared deviation (RMSD) [8] of the full molecule, estimated after alignment. Note that using just the backbone RMSD may be

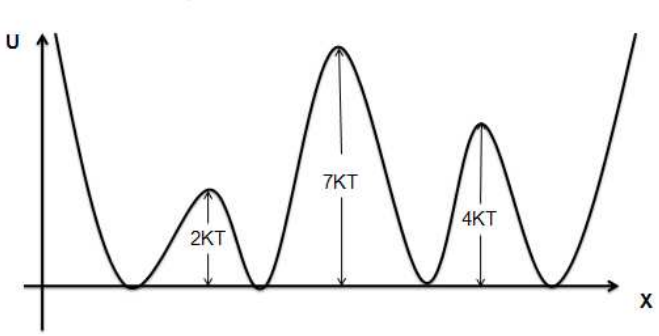

Figure 12: A one–dimensional potential energy landscape with four basins separated by three barriers.

a poor distance metric for peptides as it ignores side chain kinetics, which could assign configurations in wrong bins. However, other metrics may prove useful.

After reference structures are selected, we decompose the whole space into bins based on a Voronoi construction. That is, for each configuration, we calculate the RMSD of this configuration to each of the m reference structures. We assign the configuration to the bin associated with the reference structure, with which the configuration has the smallest RMSD.

3. Calculation of rates among bins and bin combination

We compute the mean first passage time (MFPT) from each bin, i, to every other bin, j, using a continuous dynamical trajectory or a set of trajectories. The rate from bin i to bin j is the inverse of that MFPT. In general, the rate from bin i to bin j is not the same as the rate from bin j to bin i – and we take a linear average of these two rates to define the unique effective rate between bin i and bin j, k_{ij}^{eff}. The effective rates are then used to construct a hierarchy of states.

4. Hierarchy

We construct a hierarchy of states by combining bins together if all pairs of rates k_{ij}^{eff} exceed a cutoff, k_c. The cutoff is then decreased. We start with $k_c = 1/\min(\text{MFPT})$ and progressively decrease k_c (or, equivalently, increase the transition time cutoff). With a decrease in k_c, more bins are combined resulting fewer states. Ultimately all bins are combined if transitions among all bin pairs are present in the trajectories which are analyzed.

The rule of unanimity – the requirement for fast transitions among *all* bin pairs in a state – is important for ESS estimation. In physical terms, it prevents a bin which "straddles" two states from combining with bins on both "sides" of the straddled barrier (until a suitably low k_c is employed). In turn, this absence of straddling prevents anomalous ESS estimates.

We note that the hierarchical picture can be significantly affected by the time interval between "snapshots" underlying the MFPT calculations. For example, although a trajectory may have a low likelihood (hence a low rate) to cross over the $2k_BT$ barrier in Fig. 12 in time τ_1, it may easily cross that barrier for a long enough time interval, τ_2. Thus, a hierarchical picture at the lowest level can differentiate the two left states of Fig. 12 if the rates are computed from the dynamic trajectory with snapshots at every τ_1 interval. On the other hand, if the rates are computed using the τ_2 interval, $2k_BT$ barrier cannot be resolved at the lowest hierarchical level. As an extreme case, if the interval between snapshots is longer than the largest correlation time in the system, then the rates to bin i from any other bin is simply proportional to the equilibrium population of bin i – and the application of the procedure described above is not appropriate.

Fig. 13 and 14 show the physical states in hierarchical description for dileucine and butane, respectively. Both start with $m = 20$ initial bins, and combine all the way to a single state. The effective sample size is calculated from the two-state level of the hierarchy as described in Sec. IV.B.

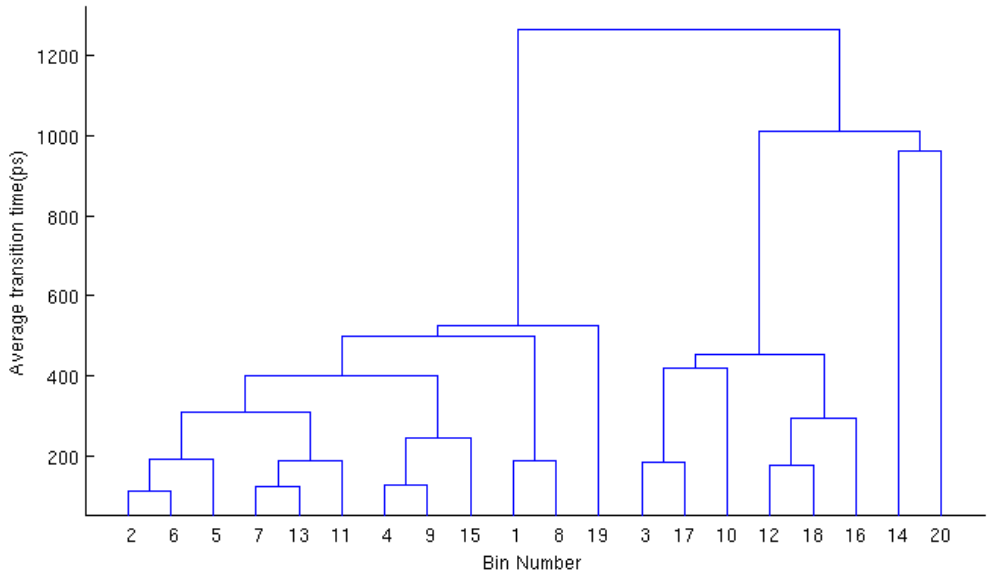

Figure 13: Hierarchical physical states for dileucine shown via the average transition time required for transition among bin pairs. Bin pairs that combine "faster" (*i.e.*, have shorter transition time) are combined at a lower level of the hierarchy.

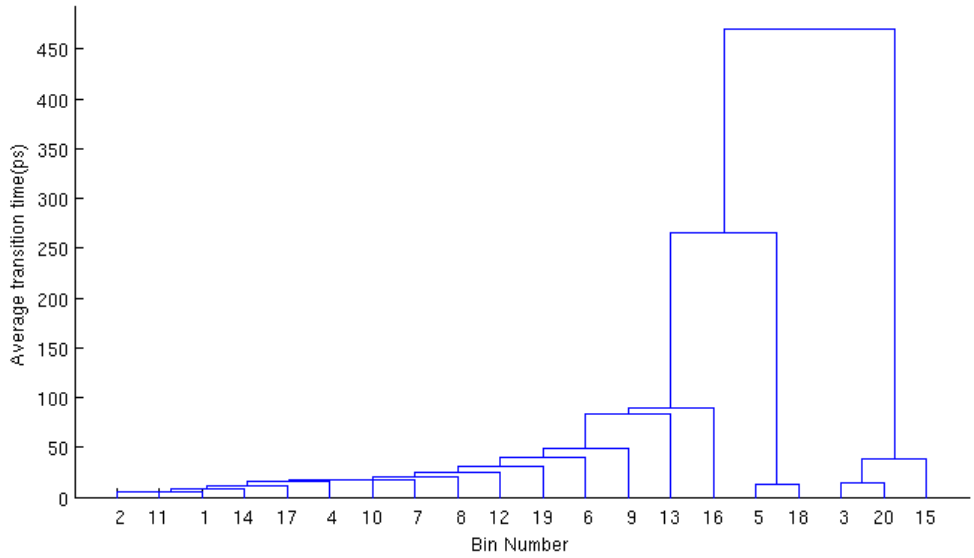

Figure 14: Hierarchical physical states for butane shown via the average transition time $(1/k_{ij}^{\text{eff}})$required for transition among bin pairs. Bin pairs that combine "faster" (*i.e.*, have shorter transition time) are combined at a lower level of the hierarchy.

C. PREVIOUS METHOD: FIND PHYSICAL STATES BY POPULATION VARIANCE

It is noteworthy that the first method employed during my research was not to combine bins based on rates. Rather, we combined the bins purely based on population variance. This method has not been published, but it also is effective to estimate the ESS. Even though both methods combine neighboring bins, they have different procedure and idea.

The variance-based method can be understood qualitatively using the example of a dynamical trajectory and two bins with minimal barrier between them. Consider neighboring bins i and j, and suppose the trajectory is in bin i at time t_1 in a dynamical simulation. The probability for the configuration in bin j is higher than other bins far away from bin i at time $t_1 + \delta t$. The combination of bin i and j will increase the suitably scaled variance of population, and thus decrease the sample size. It is similar to the two-state example we discussed in Sec. IV.D.2. In ideal sampling, the sample size estimated by each bin is reliable because there are no correlations among bins. In other words, the variance calculated in each bin from multiple simulations provides information to infer the correct timescale in ideal sampling. While in correlated trajectories, which are the most cases in biological system sampling, the configurations make the population variance reflect the convolution of fast and slow timescales (motions inside physical states and among physical states). Since we are looking for the slow timescale, neighbour bins should be combined in order to avoid the effect from fast timescale. The moral of this method is that combination of correlated bins will capture physical states, that are separated by slow time scale. The procedure could be described as

I. A reference configuration i is picked at random from the trajectory.

II. The distance of the configuration i to to all remaining configurations in the trajectory is then computed, based on an appropriate metric discussed later.

III. The configurations are sorted according to distance, and the closest N/m configurations are removed.

IV. Steps 1–3 are repeated $m-1$ times on the progressively smaller set of remaining configurations, resulting in a total of m reference configurations.

V. Calculate fractional population in each bin for all independent simulations, and find mean and variance in each bin, then calculate sample size sizes estimated in each bin

VI. Try combining all the bins to bin j that gives smallest sample size.

VII. Combine j with k, which gives smallest sample size.

VIII. Keep repeating until N^{eff} stops decreasing.

IX. Keep the same combination criteria for remaining bins until all bins are combined.

X. Pick the smallest sample size among those bins.

The discovery of physical states is carried out by sample size estimation. In other words, this method relies on the sample size in each bin (composite bins). Note that it is not necessary for the number of physical states we get from this method to be equal to two. In cases where the number of physical states is more than two, each state will have its own sample size. We will take the minimum as I discussed in Sec. IV.B.

Actually, the variance method is related to the rate method. To make it easier to understand, we can connect the two methods in this way. The population variance is related to the sample size, which is deduced from Eq. (IV.3). On the other hand, the bigger the sample size, the shorter the correlation time. Systems with short correlation time (slow time scale) need to wait shorter time to be lose memory, and thus have lower energy barriers. The states separated by low energy barrier will yield high rate. Therefore, the population variance and transition rate share the same physical insights, and both of them should work in discovery of physical states.

D. COMPARISON OF PHYSICAL STATES FROM DIFFERENT BIN SETS

To answer the question "Are physical states found by different bin sets the same?", we need to have some quantitative measurement of physicals states. Since we start from different bin sets, the physical states rarely are exactly the same. For example, we can find three states for butane systems. Let us say state A1, A2 and A3 from bin set 1, and state B1, B2, and B3 from bin set 2. It is possible that state A1 and state B1 are same region of configuration space, state A2 and B2 are in same region, and state A3 and B3 are in this

region. It could be another way around *,e.g.,* state A1 and B2 are same region, A2 and B3 are same region and A3 and B1 are same region. As long as the configuration space is divided by similar region by different independent analysis, we could conclude that the method give good estimation of physical states. We need to find a way to compare two sets of physical states from two independent analyses. So we first define a fractional population matrix based on two analyses. The states from first analysis are labeled as A1, A2,$\cdots$ and so on , while states from second analysis are labeled as B1, B2 $\cdots$ and so on. Each element of the matrix is defined as

$$T_{ij} = \frac{\text{the number of configurations in both Ai and Bj}}{\text{the total number of configurations in the trajectory}} \tag{V.1}$$

Elements with big fractional population indicate that the two states are very close with each other. For example, one analysis gives three states (A1, A2, A3), while the other one also gives three states (B1, B2, B3), the elements T_{11},T_{12} and T_{13} are 0.05, 0.1 and 0.4 respectively. The A1 has most shared regions with B3 among B1, B2 and B3. So A1 and B3 should be the same region. We can check the other states based on the same criteria – pick the biggest population among elements in each row. The quantitative measurement of physical states could rely on two variables : *Similarity* and *Overlap*.

$$Similarity = \text{sum of biggest fractional population in states from another analysis} \tag{V.2}$$

The *Similarity* gives the ratio of number of configurations that in the same states from both analysis out of the total number of configuration in the trajectory under the assumption that two states from two analyses with biggest fraction population are in same region of the configuration space. As we mentioned above, each state will have its own estimate on sample size, the sample size we pick is the smallest one. It is also very important to locate the physical state that gives the sample size. Thus, we need to compare overlap between the two states determining the sample sizes from two analyses. It should be defined as

$$Overlap = \frac{\text{fractional population intersection of the two states determining sample size}}{\text{fractional population union of the two states determining sample size}} \tag{V.3}$$

Butane	B1	B2	B3
A1	12.42%	0	0
A2	0	9.62%	0
A3	0	0	77.32%

Table 8: Fractional population in two set of physical states from two set bins for butane

The numerator is the shared region(intersction) of the two states, say A1 and B2, while the denominator is the sum (union) of the two states. The *Overlap*, unlike *Similarity*, only focus on the region that gives the sample size. It is a good indicator of whether the sample size is reliable, since the sample size of certain trajectory should be the region that sampled worst. For each system , such as butane, we can always get three states, while for flexible system, like met-enkaphalin, different number of final states could be obtained. Table 8 and 9 shows the fractional counts from two different analysis.

Based on the Eq. (V.2) and Eq. (V.3), we can calculate the *Similarity* of two butane states set are 99% (12.42% + 9.63% + 77.32%). The sample size is determined by A1 and B1, so the *Overlap* is 100% ($\frac{12.42\%+0+0}{12.42\%+0+0+0+0}$). While met-enkaphalin, the worst case, gives 86% similarity and 74% overlap. We are showing the best case and worst case among the four systems. The states we found are pretty similar, more importantly, good enough to get the sample size.

Met-enkaphalin	B1	B2	B3	B4
A1	4.5%	29.05%	16.95%	1.88%
A2	38.44%	6.06%	2.93%	0.18%

Table 9: Fractional population in two set of physical states from two set bins for met-enkaphalin

E. DISCUSSION

I demonstrate two methods to discovery physical states. The first one (rate) is to be published, while the second one (variance) is only discussed in this thesis. Both methods work well and give good estimation of ESS, but the first one is easier to understand and more fundamental than the second one and could generate full hierarchy of states landscape. The second method usually gives more than two states, and we could not the see the hierarchy. We define two variables *similarity* and *overlap*, mainly to test the accuracy of physical states. It shows that the physical states could be adequately discovered even in worst case (Met-enkaphalin).

VI. CONCLUSION AND OUTLOOK

A. WHAT HAS BEEN ACCOMPLISHED

With the continuous increase in computer power and technology, simulations of large systems at longer time-scales are becoming more feasible. However, the complexity of the free energy surface of a protein has caused many difficulties to sufficiently sample the conformational space using classical and standard molecular dynamics or Monte Carlo. The need for the development of more sophisticated techniques capable of crossing large free energy barriers has become increasingly more evident. Improvement and development of algorithms capable of sampling the entire conformational space of large protein complexes is also necessary in order to study dynamics of protein motion. Thus, the development of advanced sampling methods is crucial. As a related point, assessment of sampling efficiency is becoming more and more necessary and important. A standard and universal method is needed to test and evaluate the rising new complicated algorithms. In addition, the sampling quality and convergence of the system can help determine the statistical significance of observed results.

In this thesis I introduce how to apply the developed polymer-growth based algorithm to equilibrium sampling of several peptides systems at atomistic level. This should be the first version of library-based applications of polmer-growth algorithms in equilibrium sampling of biological systems with all atoms in implicit solvent. The statistical efficiency analysis show that this algorithm can obtain remarkable efficiency for systems for larger peptides (up to Ace-$(Ala)_{16}$-Nme). All the systems studied in this thesis could be generated in less than one minute of single CPU time, which is more than thousand times faster than standard Langevin dynamics.

I proceeded to apply this growth algorithm and extended Ytreberg and Zuckerman's

earlier work on computation of absolute free energies for molecular systems [97]. I took advantage of the pre-calculated fragment libraries, internal energy terms within fragments, as well as the interaction between bonded fragments, which makes the calculation much faster. In terms of methods, I staged the free energy differences into multiple steps in order to use the pre-generated libraries. I tested three systems — the alanine monomer, dimer, and tetramer — and obtained extremely precise free energies, with fluctuations $\ll 1$ kcal/mole. The calculations only costs less than an hour of single-processor computer time. Again, the speed results from employing pre-calculated libraries and interactions tables.

In order to show the speed and efficiency of the algorithm and provide a standard measurement on different algorithms, I have contributed to the development of a new method to assess the quality of molecular simulation trajectories – effective sample size. The new approach improves on the time correlation method [26], in several ways. The method is very objective. The sample size is what you calculate, whereas the time correlation time method is subjective and only provides ranges. More importantly, the method works for both dynamic and non-dynamic algorithms. As long as we have the information of physical states, we can estimate the effective sample size accurately and rapidly. Another feature of the new procedure is that it is applicable to discontinuous trajectories as well. We also demonstrated that the procedure is not very sensitive to the precise definitions of physical states. We tested systems ranging from discrete toy models to an all-atom treatment of rhodopsin, and got good agreement with correlation time analysis. It is also a very powerful tool to test the convergence of a simulation. In cases where ESS is estimated to be less than 10, caution should be used since the chance of inadequate sampling is very high.

To supplement the estimation of the effective sample size, I also contributed to the development of a simple procedure for the automated determination of physical states, which is based on previous work [132]. This procedure yields, in a natural way, a hierarchical picture of the configurational space, based on transition rates between regions of configuration space. I applied different means, and finally understanding the nature of the physical states, described it in a hierarchical way. The method based on transition rates and a second approach based on population variances both work for the definition of physical states. The transition rate approach has been presented in a manuscript because it is easy to explain

and apply.

There is related software available on the website (www.ccbb.pitt.edu/Zuckerman), for statistical libraries of amino-acid and capping-group fragments, sample size calculation, and physical states discovery. They are all easy to use with free download.

B. OUTLOOK

Despite the advance described, much needs to be done. The considerable speed of the growth calculations can be attributed to the use of pre-generated libraries. But the method is not well suited to very large systems. The configurations saved at an early stage may not be very helpful in a later stage. Thus, the current algorithm has limitation on the system size. One future improvement could be the "self-bias" reweighting, which assigns bigger weight on the configurations with more weight in the future steps (based on a preliminary simulation). Relaxation at each growth stage via canonical sampling may be another option help to further improve speed and efficiency.

The absolute free energy determination has proved to be very accurate and fast. A future application of potential importance is the estimation of binding affinities of small molecules to proteins. That could be very useful in drug design, where the interactions of ligands and proteins play a very important role. The methods reported here are very fast and easy to check and is very suitable for generating libraries for any kinds of small molecules.

The sample size project is a relatively complete one. The new method can analyze the effective sample size very fast based on the knowledge of physical states. There is some room to improve the speed. We are currently using (RMSD) as a "distance" measurement, but there is a more rapid alternative dRMSD. The dRMSD measure requires only simple calculation of distances between atoms without alignment and therefore should work effectively for large systems. Another improvement could be the check of sample size at each hierarchical stage to avoid the sample size increase in later stage. The variance method could avoid this potential problem. Also a population cutoff should be applied, since sample size from low-population regions is not very reliable; by definition, such regions are rarely visited and

arguably not very important.

BIBLIOGRAPHY

[1] F. C. Bernstein *et al.*, J Mol Biol **112**, 535 (1977).

[2] D. Frenkel and B. Smit, *Understanding Molecular Simulation: From Algorithms to Applications*, 2nd ed. (Academic Pr, 2001).

[3] H. C. Berg, *Random Walks in Biology*, 1st ed. (Princeton Univ Pr, 1993).

[4] A. Grossfield, P. Ren, and J. W. Ponder, J Am Chem Soc **125**, 15671 (2003).

[5] J. W. Pitera and W. F. van Gunsteren, J Am Chem Soc **123**, 3163 (2001).

[6] E. Rhoades, E. Gussakovsky, and G. Haran, Proc Natl Acad Sci U S A **100**, 3197 (2003).

[7] B. Alberts *et al.*, *Essential Cell Biology*, 2nd ed. (Garland Science/Taylor & Francis Group, 2003).

[8] A. R. Leach, *Molecular Modelling: Principles and applications* (Andrew R. Leachc, 2001).

[9] R. Parr, Annu. Rev. Phys. Chem. **34**, 631 (1983).

[10] U. Burkert and N. L. Allinger, *Molecular Mechanics* (American chemical society, 1982).

[11] S. J. Marrink, A. H. de Vries, and A. E. Mark, J. Phys. Chem. B **108**, 750 (2004).

[12] J. C. Shelley, M. Y. Shelley, R. C. Reeder, S. Bandyopadhyay, and M. L. Klein, J. Phys. Chem. B **105**, 4464 (2001).

[13] A. D. MacKerell *et al.*, J. Phys. Chem. B **102**, 3586 (1998).

[14] W. D. Cornell *et al.*, J. Am. Chem. Soc. **117**, 5179 (1995).

[15] W. L. Jorgensen, D. S. Maxwell, and J. Tirado-Rives, J. Am. Chem. Soc. **118**, 11225 (1996).

[16] C. Oostenbrink, A. Villa, A. E. Mark, and W. F. van Gunsteren, J Comput Chem **25**, 1656 (2004).

[17] P. M. Morse, Phys. Rev. **34**, 57 (1929).

[18] L. Verlet, Phys. Rev. **159**, 98 (1967).

[19] R. Hockney, S. Goel, and J. Eastwood, J. Comput. Phys **14**, 148 (1974).

[20] W. Swope, H. Andersen, P. Berens, and K. Wilson, J. Chem. Phys. **76**, 637 (1982).

[21] D. Beeman, J. Compu. Phys. **20**, 130 (1976).

[22] P. Langevin, C. R. Hebd. Seances Acad Sci. **146**, 530533 (1908).

[23] H. Flyvbjerg and H. G. Petersen, J. Chem. Phys. **91**, 461 (1989).

[24] B. Hess, PHYS. REV. E **65**, 031910 (2002).

[25] E. Lyman and D. M. Zuckerman, Biophys. J. **91**, 164 (2006).

[26] E. Lyman and D. M. Zuckerman, J Phys Chem B **111**, 12876 (2007).

[27] E. M. Marcotte *et al.*, Science **285**, 751 (1999).

[28] D. J. Wales, Int. Rev. Phys. Chem. **25**, 237 (2006).

[29] A. Grossfield and D. M. Zuckerman, Annu. Rep. Comput. Chem. **5**, 23 (2009).

[30] R. H. Swendsen and J.-S. Wang, Phys. Rev. Lett. **57**, 2607 (1986).

[31] Y. Sugita and Y. Okamoto, Chem. Phys. Lett. **314**, 141 (1999).

[32] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, J. Chem. Phys. **21**, 1087 (1953).

[33] A. B. Mamonov, D. Bhatt, D. J. Cashman, Y. Ding, and D. M. Zuckerman, J Phys Chem B **113**, 10891 (2009).

[34] J. S. Liu, *Monte Carlo Strategies in Scientific Computing* (Springer, New York, 2004).

[35] Y. Ding, A. B. Mamonov, and D. M. Zuckerman, J. Phys. Chem. B (2010).

[36] E. W. Montroll, J. Chem. Phys. **18**, 734 (1950).

[37] J. Hammersley and K. Morton, J. Roy. Stat. Soc. B **5**, 23 (1954).

[38] F. Wall and L. Hiller, Ann. Rev. Phys. Chem. **5**, 267 (1954).

[39] F. Wall, L. Hiller, and D. Wheeler, J. Chem. Phys. **22**, 1036 (1954).

[40] M. N. Rosenbluth and A. W. Rosenbluth, J. Chem. Phys. **23**, 356 (1955).

[41] F. T. Wall, R. J. Rubin, and L. M. Isaacson, J. Chem. Phys. **27**, 186 (1957).

[42] F. T. Wall and J. J. Erpenbeck, J. Chem. Phys. **30**, 634 (1959).

[43] Z. Alexandrowicz, J. Chem. Phys. **51**, 561 (1969).

[44] H. Meirovich, J. Phys. A: Math. Gen. **15**, L735 (1982).

[45] H. Meirovich, Phys. Rev. A **32**, 3699 (1985).

[46] T. Garel and H. Orland, J. Phys. A: Math. Gen. **23**, L621 (1990).

[47] P. Grassberger, Phys. Rev. E **56**, 3682 (1997).

[48] P. Grassberger, Comput. Phys. Commun. **147**, 64 (2002).

[49] U. Bastolla, H. Frauenkron, E. Gerstner, P. Grassberger, and W. Nadler, Proteins **32**, 52 (1998).

[50] J. L. Zhang and J. S. Liu, J. Chem. Phys. **117**, 3492 (2002).

[51] J. Zhang, M. Lin, R. Chen, J. Liang, and J. S. Liu, PROTEINS **66**, 61 (2007).

[52] T. Garel, J. C. Niel, H. Orland, J. Smith, and B. Velikson, J. Chem. Phys **88**, 2479 (1991).

[53] B. Velikson, T. Garel, J. C. Niel, H. Orland, and J. C. Smith, J. Comput. Chem. **13**, 1216 (1992).

[54] J. Bascle, T. Garel, H. Orland, and B. Velikson, Biopolymers **33**, 1843 (1993).

[55] A. B. Mamonov, X. Zhang, and D. Zuckerman, J. Comp. Chem. **In press** (2010).

[56] B. A. Berg and T. Neuhaus, Phys. Lett. B **267**, 249 (1991).

[57] U. H. E. Hansmann, Chem. Phys. Lett. **281**, 140 (1997).

[58] N. Nakajima, H. Nakajima, and J. Kidera, J. Phys. Lett. B **101**, 817 (1997).

[59] E. Lyman and D. Zuckerman, J. Chem. Theory Comput. **2**, 656 (2006).

[60] L. Nymeyer, J. Chem. Theory Comput. **4**, 626 (2008).

[61] R. Denschlag, M. Lingenheil, and P. Tavan, Chem. Phys. Lett. **458**, 244 (2008).

[62] E. Lyman, F. M. Ytreberg, and D. M. Zuckerman, Phys. Rev. Lett. **96**, 028105 (2006).

[63] E. Lyman, J. Pfaendtner, and G. Voth, Biophys. J. **95**, 4183 (2008).

[64] M. Winger, D. Trzesniak, R. Baron, and W. wan Gunsteren, Phys. Chem. Chem. Phys. **11**, 1934 (2009).

[65] J. Zhang, S. Kou, and J. Liu, J. Chem. Phys. **126**, 225101 (2007).

[66] C. A. Rohl, C. E. M. Strauss, K. M. S. Misura, and D. Baker, Method Enzymol. **383**, 66 (2004).

[67] A. Shehu, C. Clementi, and L. Kavraki, Proteins **65**, 164 (2006).

[68] A. Shehu, C. Clementi, and L. Kavraki, Algorithmica **48**, 303 (2007).

[69] X. Zhang, A. B. Mamonov, and D. M. Zuckerman, J. Comp. Chem **30**, 1680 (2009).

[70] A. Ferrenberg and R. Swendsen, Phys. Rev. Lett. **61**, 2635 (1988).

[71] D. Qiu, P. S. Shenkin, F. P. Hollinger, and W. C. Still, J Phys. Chem. A **101**, 3005 (1997).

[72] P. Fearnhead and P. Clifford, J. R. Stat. Soc. B **65**, 887 (2003).

[73] W. Yang, H. Nymeyer, H.X.Zhou, B. Berg, and R. Bruschweiler, J. Comput. Chem. **29**, 668 (2008).

[74] J. Hegler, J. Latzer, A. Shehu, C. Clementi, and P. Wolynes, Proc. Natl. Acad. Sci. U. S. A. **106**, 15302 (2009).

[75] S. Ozkan, G. Wu, J. Chodera, and K. Dill, Proc. Natl. Acad. Sci. U. S. A. **104**, 11987 (2007).

[76] T. Rathinavelan and W. Im, J. Comput. Chem. **29**, 1640 (2008).

[77] G. Voronoi, Journal fr die Reine und Angewandte Mathematik. **133**, 97 (1907).

[78] A. Miranker and M. Karplus, Proteins **11**, 29 (1991).

[79] M. D. Macedonia and E. J. Maginn, Mol. Phys. **96**, 1375 (1999).

[80] F. Wall, R. Rubin, and L. Isaacson, J. Chem. Phys. **27**, 186 (1957).

[81] C. Karney, J. Mol. Graph Model **25**, 595 (2007).

[82] K. Nitadori, J. Makino, and P. Hut, New Astronomy **12**, 169 (2006).

[83] G. N. RAMACHANDRAN, C. RAMAKRISHNAN, and V. SASISEKHARAN, J Mol Biol **7**, 95 (1963).

[84] N. Sewald and H. Jakubke, *Peptide: Chemistry and Biology* (Wiley-VCH, 2009).

[85] L. Otvos, *Peptide-based drug design: here and now* (Springer, 2008).

[86] D. Frenkel and B. Smit, *Understanding Molecular Simulations* (Academic Press, San Diego, 2002).

[87] G. Chikenji, Y. Fujitsuka, and S. Takada, J. Chem. Phys. **119**, 6895 (2003).

[88] R. M. Neal, Stat. comput. **11**, 125 (2001).

[89] E. Lyman and D. M. Zuckerman, J. Chem. Phys. **127**, 065101 (2007).

[90] E. Lyman and D. Zuckerman, J. Chem. Phys. **130**, 8 (2009).

[91] D. Bhatt and D. Zuckerman, J. Phys. Chem. B **(submitted)** (2009).

[92] W. L. Jorgensen, Science **303**, 1813 (2004).

[93] S. B. Singh, Ajay, D. E. Wemmer, and P. A. Kollman, Proc. Natl. Acad. Sci **91**, 7673 (1994).

[94] Y. Deng and B. Roux, J. Chem. Theory Comput. **2**, 1255 (2006).

[95] C. Jarzynski, Phys. Rev. Lett. **78**, 2690 (1997).

[96] C. Jarzynski, Phys. Rev. E **56**, 5018 (1997).

[97] F. M. Ytreberg and D. M. Zuckerman, J Chem Phys **124**, 104105 (2006).

[98] D. Frenkel and B. Smit, *Understanding Molecular Simulation* (Academic Press, San Diego, 1996).

[99] J. P. Stoessel and P. Nowak, Macromol. **23**, 1961 (1990).

[100] L. Huang and D. E. Makarov, J. Chem. Phys. **124**, 64108 (2006).

[101] H. Meirovitch, J. Chem. Phys. **97**, 5803 (1992).

[102] H. Meirovitch, J. Chem. Phys. **111**, 7215 (1999).

[103] S. Cheluvaraja and H. Meirovitch, Proc. Natl. Acad. Sci. U. S. A. **101**, 9241 (2004).

[104] R. P. White and H. Meirovitch, Proc. Natl. Acad. Sci. U.S.A **101**, 9235 (2004).

[105] M. S. Head, J. A. Given, and M. K. Gilson, J. Phys. Chem. B **101**, 1609 (1997).

[106] C. Chang, P. Michael, and M. Gilson, J. Phys. Chem. B **107**, 1048 (2003).

[107] C. Chang and M. Gilson, J. Am. Chem. Soc. **126**, 13156 (2004).

[108] W. Chen, C. Chang, and M. Gilson, Biophys. J **87**, 3035 (2004).

[109] N. Go and H. Scheraga, J. Chem. Phys. **51**, 4751 (1969).

[110] M.N.Rosenbluth and A. Rosenbluth, J. Chem. Phys. **23**, 356 (1955).

[111] P. Grassberger, J. Phys. A: Math. Gen. **26**, 2769 (1993).

[112] P. Grassberger and R. Hegger, J. Phys.- Condens. Mat. **7**, 3089 (1995).

[113] K. D. Gibson and H. A. Scheraga, J. Comput. Chem. **8**, 826 (1987).

[114] R. W. Zwanzig, J. Chem. Phys. **22**, 1420 (1954).

[115] D. A. K. Peter and T. Cummings, Mol. Phys. **92**, 973 (1997).

[116] D. A. Kofke and P. T. Cummings, Fluid Phase Equilibr. **150-151**, 41 (1998).

[117] J. W. Ponder and F. M. Richard, J. Comput. Chem. **8**, 1016 (1987).

[118] T. M. Cover and D. A. Thomas, *Elements of Information Theory,2nd edition* (Wiley, 2006).

[119] X. Zhang, D. Bhatt, and D. Zuckerman, J. Chem. Theory Comput. **In press** (2010).

[120] B. A. Berg and T. Neuhaus, Phys. Rev. Lett. **68**, 9 (1992).

[121] Y. Okamoto., J. Mol. Graph.Model. **22**, 425 (2004).

[122] J. B. Abrams and M. E. Tuckerman, J. Phys. Chem. B **112**, 1574215757 (2008).

[123] P. Ren and J. Ponder, J. Phys. Chem. B **107**, 5933 (2003).

[124] G. Lamoureux, A. Mackerell, and B. Roux, J. Chem. Phys. **119**, 5185 (2003).

[125] B. Keller, X. Daura, and W. F. van Gunsteren, J. Chem. Phys. **132**, 074110 (2010).

[126] L. E. Reich, *A Modern Course in Statistical Physics* (Springer, 2009).

[127] S. Wenzel and W. Janke, Phys. Rev. B **79**, 014410 (2009).

[128] K. Binder and D. W. Heermann, *Monte Carlo Simulation in Statistical Physics* (Springer, Berlin, 1997).

[129] A. Grossfield, S. E. Feller, and M. C. Pitman, Proteins: Struct. Funct. Bioinf. **67**, 31 (2007).

[130] R. D. Mountain and D. Thirumalai, J. Phys. Chem **93**, 6975 (1989).

[131] R. D. Mountain and D. Thirumalai, Int. J. Mod. Phys. C **1**, 77 (1990).

[132] J. D. Chodera, N. Singhal, W. C. Swope, V. S. Pande, and K. A. Dill, J. Chem. Phys. **126**, 155101 (2007).

[133] F. Noé, I. Horenko, C. Schütte, and J. C. Smith, J. Chem. Phys. **126**, 155102 (2007).

[134] D. J. Earl and M. W. Deem, Phys. Chem. Chem. Phys. **7**, 3910 (2005).

[135] N.-V. Buchete and G. Hummer, Phys. Rev. E **77**, 030902 (2008).

[136] J. D. Chodera, W. C. Swope, J. W. Pitera, C. Seok, and K. A. Dill, J. Chem. Theory Comput. **3**, 26 (2007).

[137] X. Huang, G. R. Bowman, and V. S. Pande, J. Chem. Phys. **128**, 205106 (2008).

[138] E. Rosta and G. Hummer, J. Chem. Phys. **131**, 134104 (2009).

[139] D. M. Zuckerman, J. Phys. Chem. B **108**, 5127 (2004).

[140] J. Michel, R. D. Taylor, and J. Essex, J. Chem. Theory Comput. **2**, 732739 (2006).

[141] J. Mongan, C. Simmerling, J. A. McCammon, D. A. Case, and A. Onufriev, J. Chem. Theory Comput. **3**, 156169 (2007).

[142] A. Grossfield, S. Feller, and M. Pitman, Proc. Nat. Acad. Sci. **103**, 4888 (2006).

[143] H. Frauenfelder, S. Sligar, and P. Wolynes, Science **254**, 1598 (1991).

[144] Y. Levy, J. Jortner, and O. M. Beckera, J. Chem. Phys. **115**, 22 (2001).

[145] H. Fraunfelder, F. parak, and R. D. Young, Annu. Rev. Biophys. Biophys. Chem. **17**, 451 (1988).

[146] D. J. Wales, J. Chem. Phys. **130**, 204111 (2009).